2006

Evaluating Microsoft .NET technology: Implementation online store

Jie Dou

EVALUATING MICROSOFT .NET TECHNOLOGY: IMPLEMENTATION ONLINE STORE

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Jie Dou

June 2006

EVALUATING MICROSOFT .NET TECHNOLOGY:

IMPLEMENTATION ONLINE STORE

A Project

Presented to the

Faculty of

California State University,

San Bernardino

---

by

Jie Dou

June 2006

Approved by:

[redacted]

Dr. Ernesto Gomez — 6/7/06 Date

[redacted]

Dr. David Turner

[redacted]

Dr. Kerstin Voigt

ABSTRACT

This project is a web application implemented using Microsoft .NET Technology. The purpose is to design, develop and implement an e-commerce shopping cart system based on ASP .NET technology, and also via developing such a shopping cart system, to evaluate ASP .NET technology.

This web application is a virtual company designed by the author to handle B-to-C online business, which has two categories of clients: customer, and administrator. This project allows clients to enter the web pages based on their category level. This document not only describes the details of the project in both application mechanism and functionality, but also presents the source code and explanations.

ACKNOWLEDGMENTS

I thank the faculty of Computer Science department for giving me an opportunity to pursue my M.S. in Computer Science at California State University, San Bernardino. I express my sincere appreciation to my graduate advisor, Dr. Ernesto Gomez who offered me this project and directed me through this entire effort. I also thank my other committee members, Dr. David Turner and Dr. Kerstin Voigt for their valuable input.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Purpose of the Project

The purpose of this project is to design, develop and implement an e-commerce shopping cart system based on Microsoft .NET technology, to evaluate ASP .NET technology via developing such a shopping cart system.

This project includes those steps:

1. Set up a site with .NET technology.

In this step, we decide what technology to use to build our project. We start by putting together our basic site architecture, deciding how different parts of our application work together. We set up an IIS web server, use Visual Basic .NET and work in Visual Studio .NET IDE database environment.

2. Build product catalog into our architecture

In this step we design a database for storing a product catalog, containing categories, sub-categories and products; write the SQL and VB .NET code for accessing that data; provide a free-text search engine for that database; and create administrator section to modify catalog online.

3. Build a shopping cart

In this step, we build ASP .NET shopping cart with our desired feel and look; add database table for storing complete orders, integrate our shopping cart system.

4. Evaluate ASP .NET technology

While ASP .NET technology is very broad area and has a lot of components, we focus on evaluating these aspects:

A. The costs of building an e-commerce site with ASP .NET technology
B. Time and convenience to build an e-commerce site with ASP .NET technology
C. Technical advantages of ASP .NET in building an e-commerce site

1.2 Assumptions

This software is aimed at developers looking for approach to building a full e-commerce web site from design to develop. However, it's assumed that developers:

1. Have some knowledge of using ASP .NET with VB .NET
2. Have experience with Visual Basic .NET
3. Interest on Microsoft .NET technology

CHAPTER TWO

TECHNOLOGY

2.1 Introduction

Chapter Two consists of a discussion of technologies that were used to develop this web application, including Microsoft .NET Framework architecture, ASP .NET, language Visual Basic .NET, ADO .NET and Database Server MSDE.

2.2 .NET Framwork

.NET is the Microsoft Web services strategy to connect information, people, systems, and devices through software[1]. The .NET framework is a programming internet and Web based infrastructure created by Microsoft for building, deploying, and running applications and services that use .NET technologies, such as desktop applications and Web services. The .NET Framework contains three major parts: the Common Language Runtime (CLR), the Framework Class Library, and ASP .NET [2].

From Figure 1, we can see clearly .NET framework infrastructure has languages at the top such as VB .NET C#, VJ#, VC++ .NET; developers can develop (using any of above languages) applications such as Windows Forms, Web Form, Windows Services and XML Web Services. Bottom two layers consist of .NET Framework class library and Common

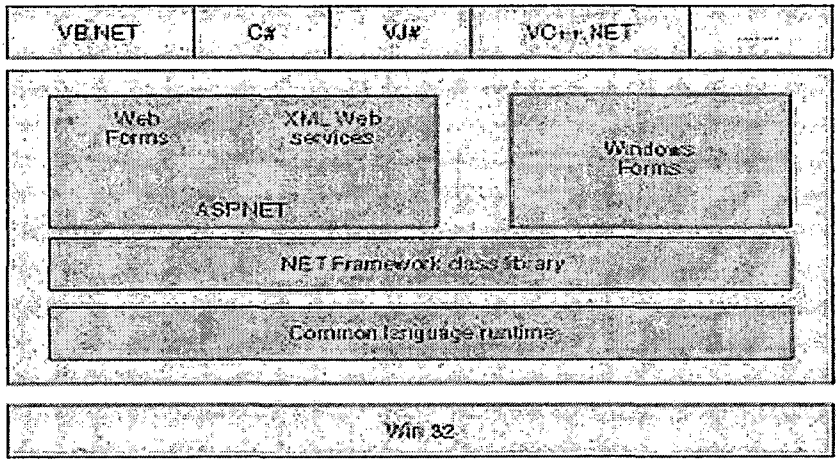

Figure 1. .NET Framework Infrastructure

Language Runtime.

2.3 ASP .NET

ASP .NET is a technology based on .NET framework for creating dynamic web applications. It provides a unified web development model that includes the services necessary for developers to build enterprise-class Web applications. Its architecture makes the ASP .NET to optimize the Windows operating. ASP .NET can author applications in any .NET compatible language, including Visual Basic .NET, C#, and J#.Additionally, the entire .NET Framework is available to any ASP .NET application.

As the Figure 2 shows, all web clients communicate with ASP .NET applications through Microsoft Internet Information Service (IIS). It also provides a programming

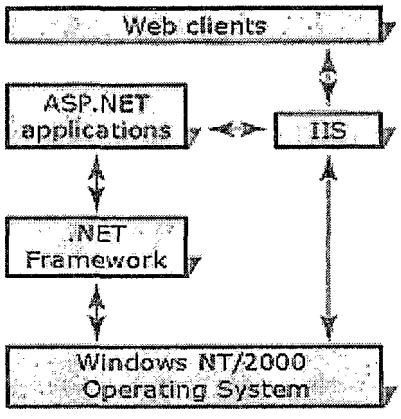

Figure 2. ASP .NET Architecture

model and infrastructure for more scalable and stable applications that help provide greater protection.

2.4 Visual Basic .NET

Visual Basic .NET is one of the languages that can be used to code the web form's logic. It combines the power of the .NET framework and the common language runtime with the productivity benefits that are the hallmark of Visual Basic. Unlike its previous version (VB6), VB .NET has added language enhancements, making it a fully object-oriented language and taking advantage of the features provided by the .NET framework.

Visual Studio .NET is by far the most powerful tool to develop .NET applications. It provides a feature-rich application execution environment, simplified development

and easy integration between different development languages. In this project, I will use Visual Studio .NET to develop ASP .NET web application.

2.5 ADO .NET

In this project, I will use Microsoft Database Engine (MSDE) as data source. The ADO .NET [3] is a technology that permits accessing a database from Visual Basic .NET code. All .NET classes that are related to database access are collectively known as ADO .NET. There are two central components of ADO .NET that accomplish this: Dataset and .NET Framework Data Provider. The Figure3 [6] illustrates the components of ADO .NET architecture.

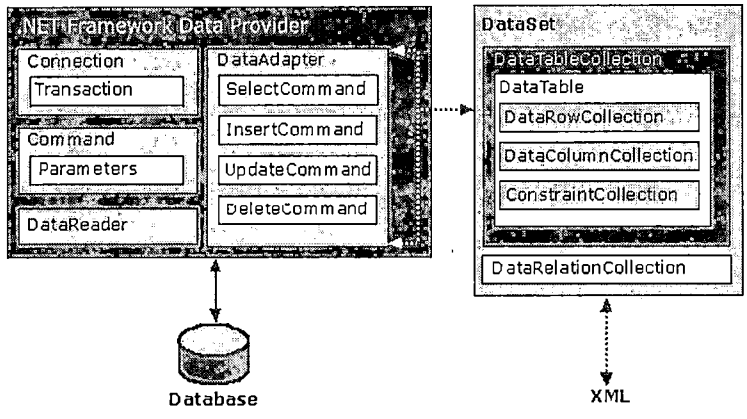

Figure 3. ADO .NET Data Architecture

The DataSet is a disconnected, in-memory representation of data. It contains a collection of one or

more DataTable objects made up of rows and columns of data. The .NET Framework Data is responsible for providing and maintaining the connection to the database. A DataProvider is [6] a set of related components that work together to provide data in an efficient and performance driven manner. ADO .NET is the most modern Microsoft data-access technology, and it can be used from any .NET language, not only with Visual Basic .NET.

2.6 Microsoft Database Engine (MSDE)

Visual Studio .NET ships with a free version of SQL server named MSDE (Microsoft Database Engine). MSDE has some performance limitations, but it is free and totally compatible with the commercial and expensive version of SQL server. That means that developers can use MSDE on their development machines and then simply transfer the database the full SQL Server versions at the client site. This is an important advantage over the old technique of using Access on development machines and then upsizing to SQL Server when moving the application to the client site.

CHAPTER THREE

PROJECT DESIGN

3.1 Introduction

Chapter Three includes project perspective, project architecture design and system design. Specially, system design discussed database design in details by E-R diagram. According to the design, the database will have 7 tables.

3.2 Product Perspective

This project is web-based, and will be integrated with .NET Framework. The hardware interface requirement is that it must run on the existing web server or local host. The software interface requirement is that it must support current versions of Internet Explorer or Netscape. The communication interface requirement is that it must support HTTP. The system will be opened 24 hours a day, 7 days per week. All actions are user initiated. No separate backup and recovery or maintenance functions are required as that is handled by system administration on the hosting server machine.

3.2.1 Software Interface

Due to the limited availability of browsers capable of using the visual Basic .NET, ASP .NET, the following browsers/platforms will be supported at the minimum:

Table 1. Software Interface

Software	System
Operating System	Windows 2000/XP Professional
Internet Browser	Microsoft IE 5.0 or above Netscape
Web Server	IIS 5.x Web server
Database	MSDE /Microsoft SQL Server 2000
ASP .NET Platform	V1.00.3705.6018
VB Compiler	V7.00.9951

3.2.2 Hardware Interface

Online store will not directly implement any hardware interfaces. All interfacing to I/O devices will be provided by the window operating system in this project.

3.2.3 Communication Interface

The communication interfaces are HTTP for general information. Private information is encrypted using SSL protocols. Communication between application and database MSDE is through ADO .NET bound in the Visual Studio .NET software.

3.2.4 Memory Constraints

There is no specific memory requirement for the customer computer. Although there is no explicit memory requirement for the web server and database server, enough memory is required to guarantee acceptable response time. 128 Mega bytes or higher are recommended.

3.2.5 Operations

Online store be visited by users and operated by administrator.

3.3 Project Architecture Design

The three-tier architecture [10] has become popular today because it answers most of the problems discussed so far by slitting an application's functionality into three logical tiers: the presentation tier, the business tier and the data tier. This project will use three-tier architecture to design the online-store.

The presentation tier [10] contains the user interface element of the site, and includes all the logic that manages the interaction between the visitor and the client's business. The presentation tier is composed of dynamic web pages in this project.

The business tier [10] (also called the middle tier) receives requests from the presentation tier and returns a

result to the presentation tier depending on the business logic it contains.

The data tier (something referred to as the database tier) is responsible for storing the application's data, and sending it to the business tier when requested.

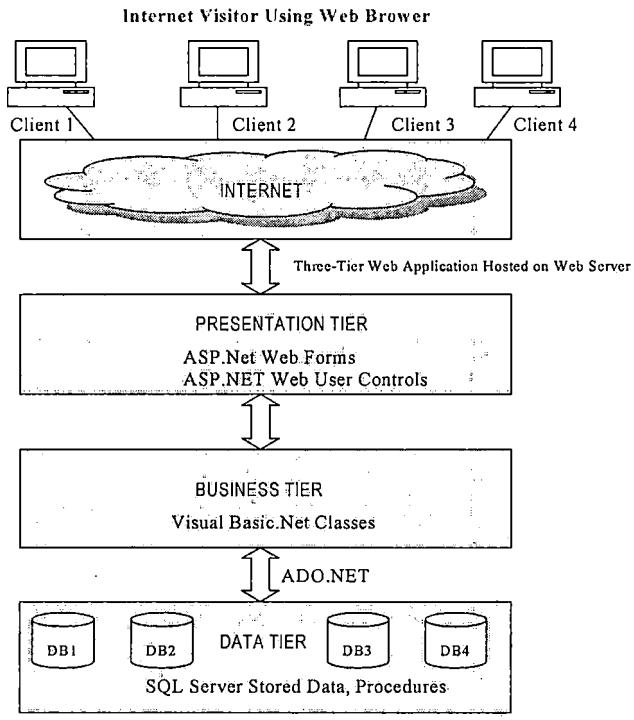

Figure 4. Project Architecture

An important constraint in the three-tier architecture model is that information must flow in sequential order between tiers. The presentation tier is only allowed to access the business tier, and never directly the data tier. The business tier is the "brain"

in the middle that communicates with the other tiers and process and coordinates all the information flow.

3.4 Project System Design

3.4.1 Use Case Diagram

This system is separated into two parts based on the two client categories. Figure 5 shows 5 principal functions for two different users: customers, administrator. Customers can register and edit their account, receive order confirmation emails, add final product to shopping cart and edit shopping cart. After administrators log in administrator account, they can maintenance customer account, process orders, and maintenance product catalog.

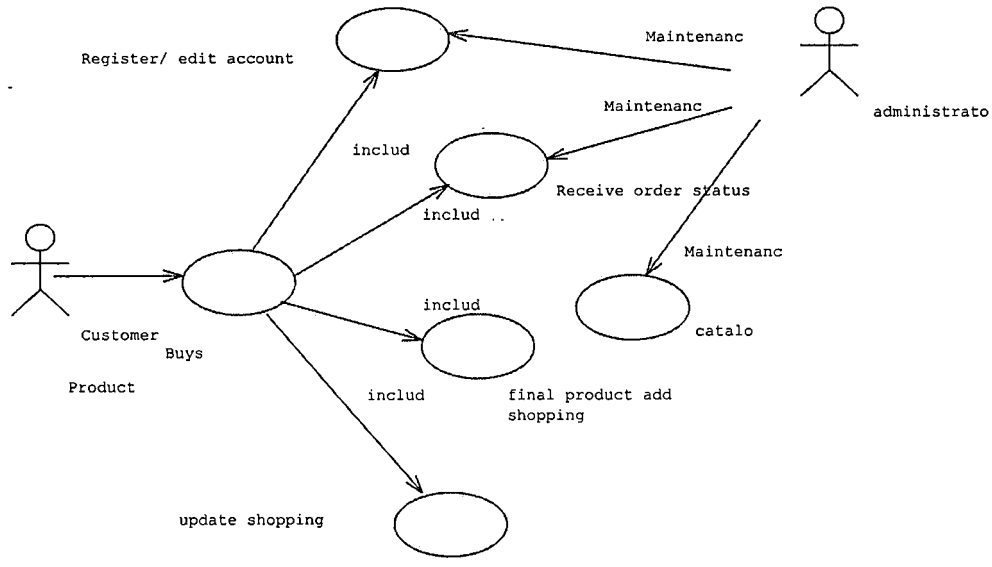

Figure 5. Use Case Diagram

3.4.2 Database Design

In this project, I assume catalog consist of department, category. Each department can have many categories, but each category can exist in only one department. Each product can exist in MORE THAN ONE category. A customer can place many orders, but one order can only be placed by one customer.

Figure 6 is the ER diagram showing the relations and the important attributes of the important entities.

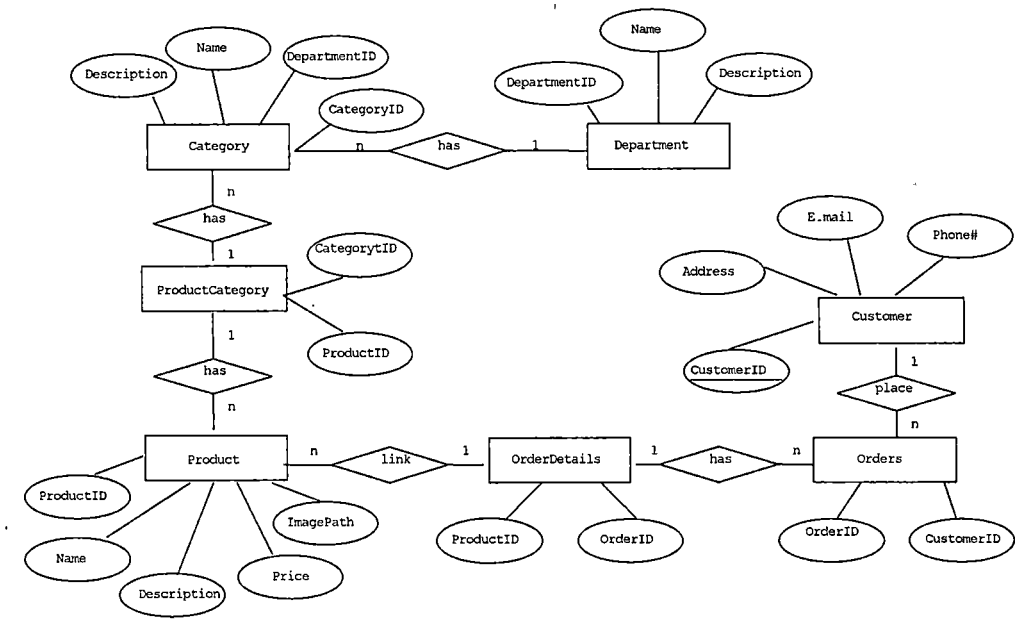

Figure 6. Database Entity Relationship Diagram

3.5 Project Security Design

In this project, we will use secure sockets layer (SSL) to proect security sentitive HTML pages such like log in page, account management pages.

Secure Sockets Layer (SSL) is a protocol developed by Netscape for transmitting private documents via the Internet. [13] SSL uses a cryptographic system that uses two keys to encrypt data - a public key known to everyone and a private or secret key known only to the recipient of the message. Both Netscape Navigator and Internet Explorer support SSL, and many Web sites use the protocol to obtain confidential user information, such as credit card numbers. By convention, URLs that require an SSL

connection start with *https:* instead of *http*:. When customers visit the website, URL begins with http://; when customers need log in account or register accout and administrator need log in account, URL switches to https://. After they finish the work, and log out account, go back to regular webpages which don't contain sentitive privacy information, then URLs swith from https:// to http again.

CHAPTER FOUR

PROJECT IMPLEMENTATION

4.1 Introduction

Included in Chapter Four was a presentation of the implementation of the project. Following sections will discuss GUI design, class design and all functions implementation.

4.2 Graphic User Interface (GUI)

The website is easy to use. The WebPages are written using ASP .NET in Visual Basic .NET Environment. All the Catalog function is in the left region of the layout page. And, the contents part is placed at the body part which is in the lower region of the layout page. The following sub sections explain the GUI.

4.2.1 Home Page

The main web page contains tree major sections. The three table cells will be filled with user controls: list of Departments, Header control, Contents control. The contents cell is the only one that changes while browsing the site. The other two cells will be filled with the same controls no matter what page is visited.

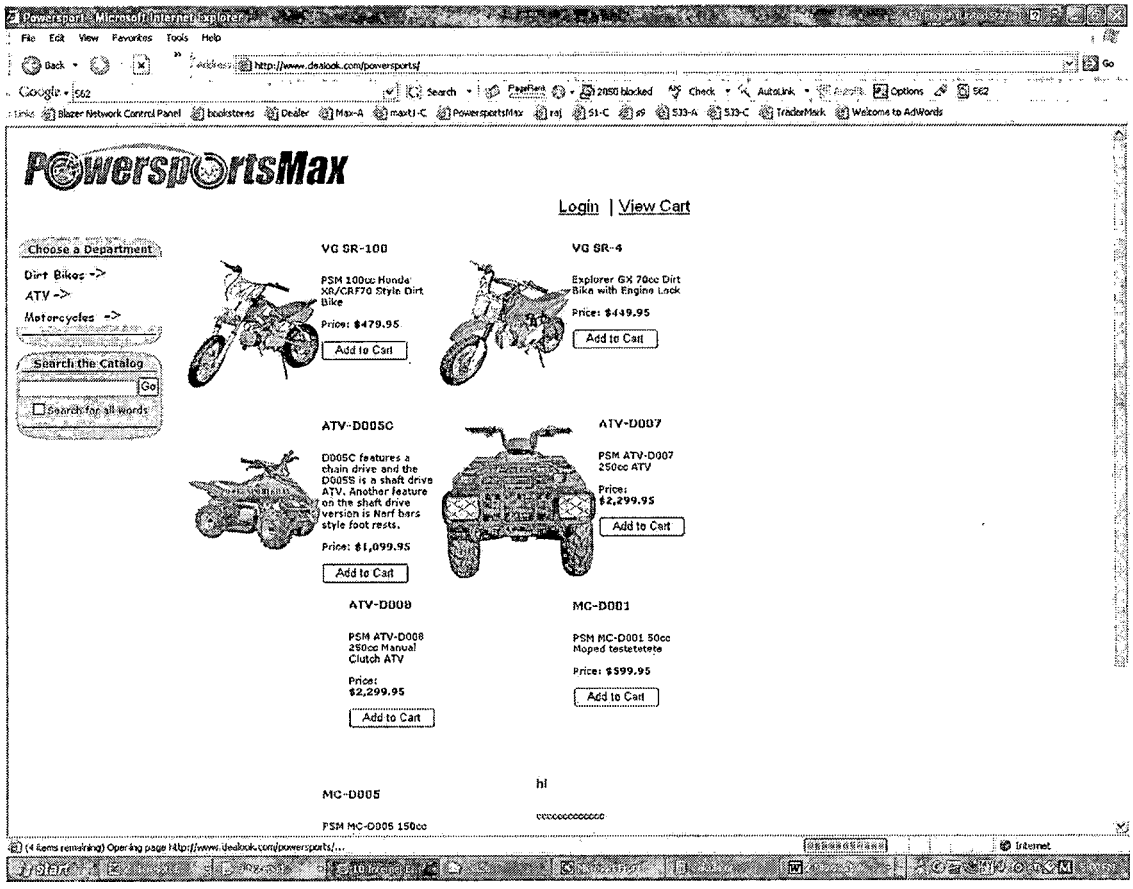

Figure 7. Store Home Page

4.2.2 Administrator Login Page

In this page administrator can log in after input user name and password.

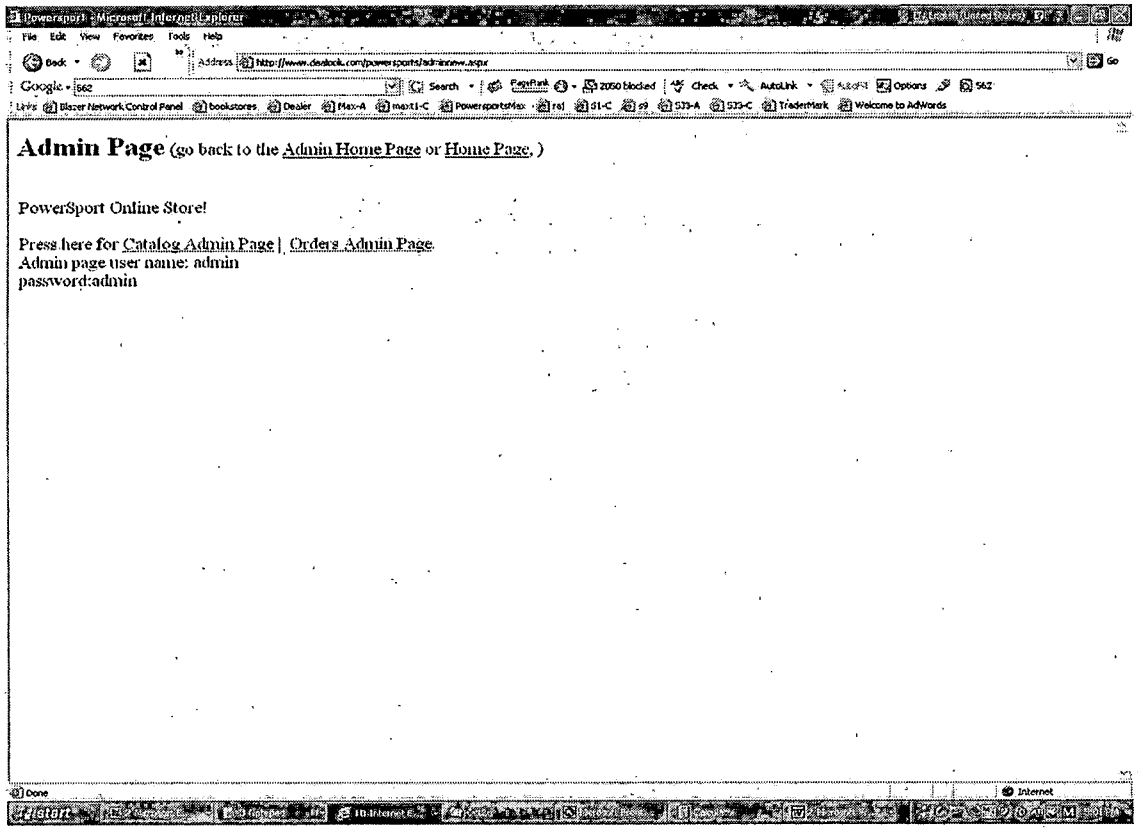

Figure 8. Administrator Login Page

4.2.3 Catalog Admin Page

In this page administrator can edit the department's name or description when click edit button, edit the categories for a specific department by click Edit Categories button, and completely remove a department from the database by clicking the Delete button.

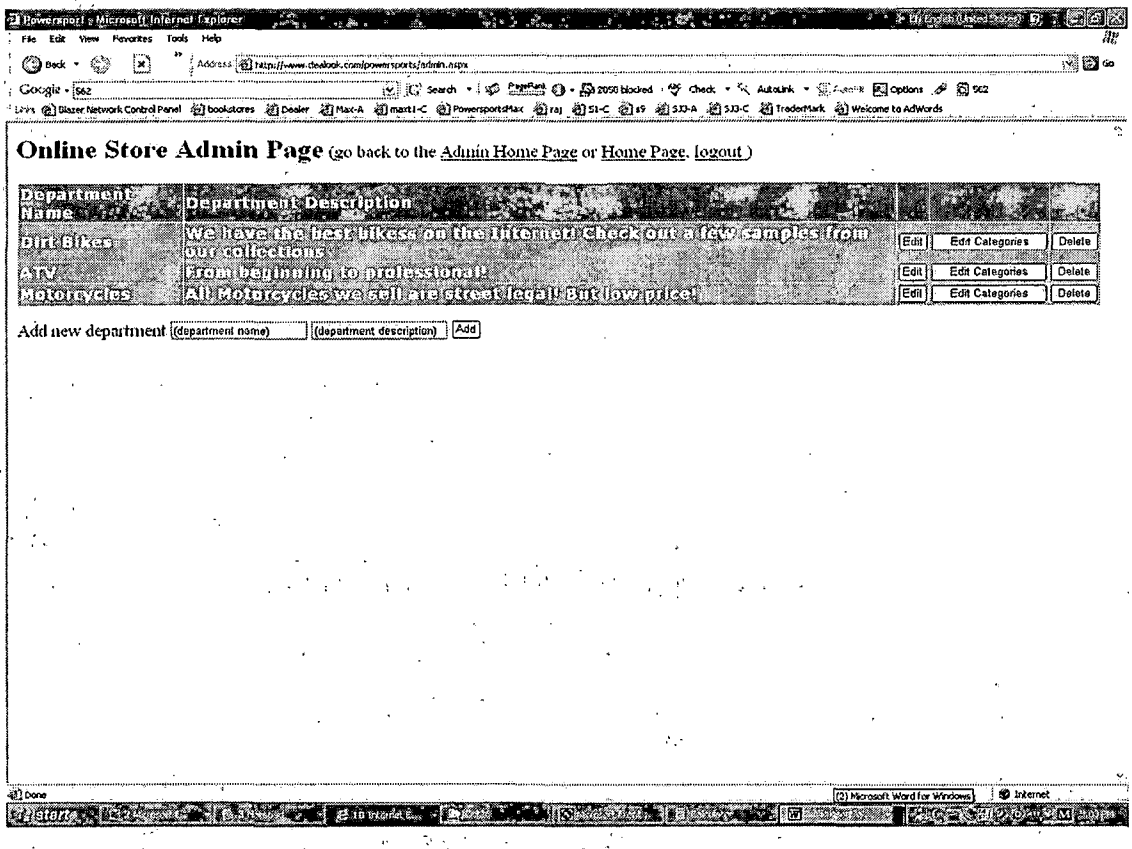

Figure 9. Department Admin Page

In Catalog admin page, administrator can edit and update catalog description and name by clicking "Edit" button; add new catalog after clicking "Add" button at the bottom of the page; manage products when clicking " Edit Products" button, and remove this catalog when clicking "Delete" button.

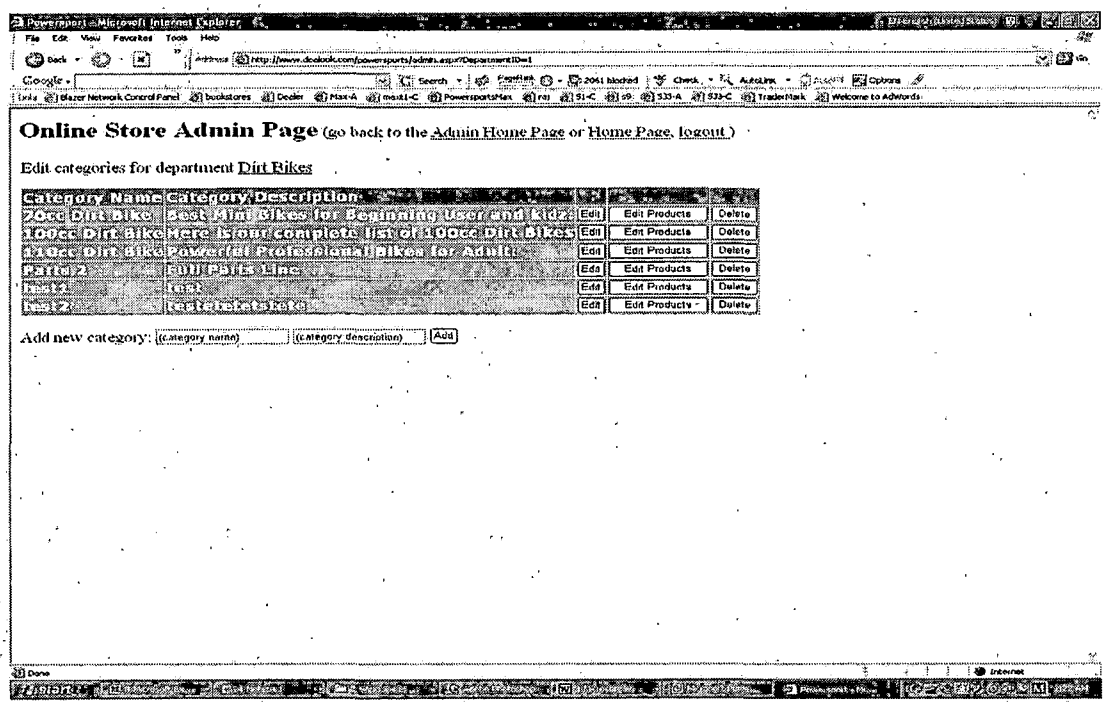

Figure 10. Catalog Admin Page

Administrator can manage products in product page. Adminstrator can edit and update product name, price, description, images and promotion when clicking "Edit" button; also can add new products and assign products to department promotion or catalog promotion section for crossing sale. After clicking "Select" button, product admin page will go to product detail page.

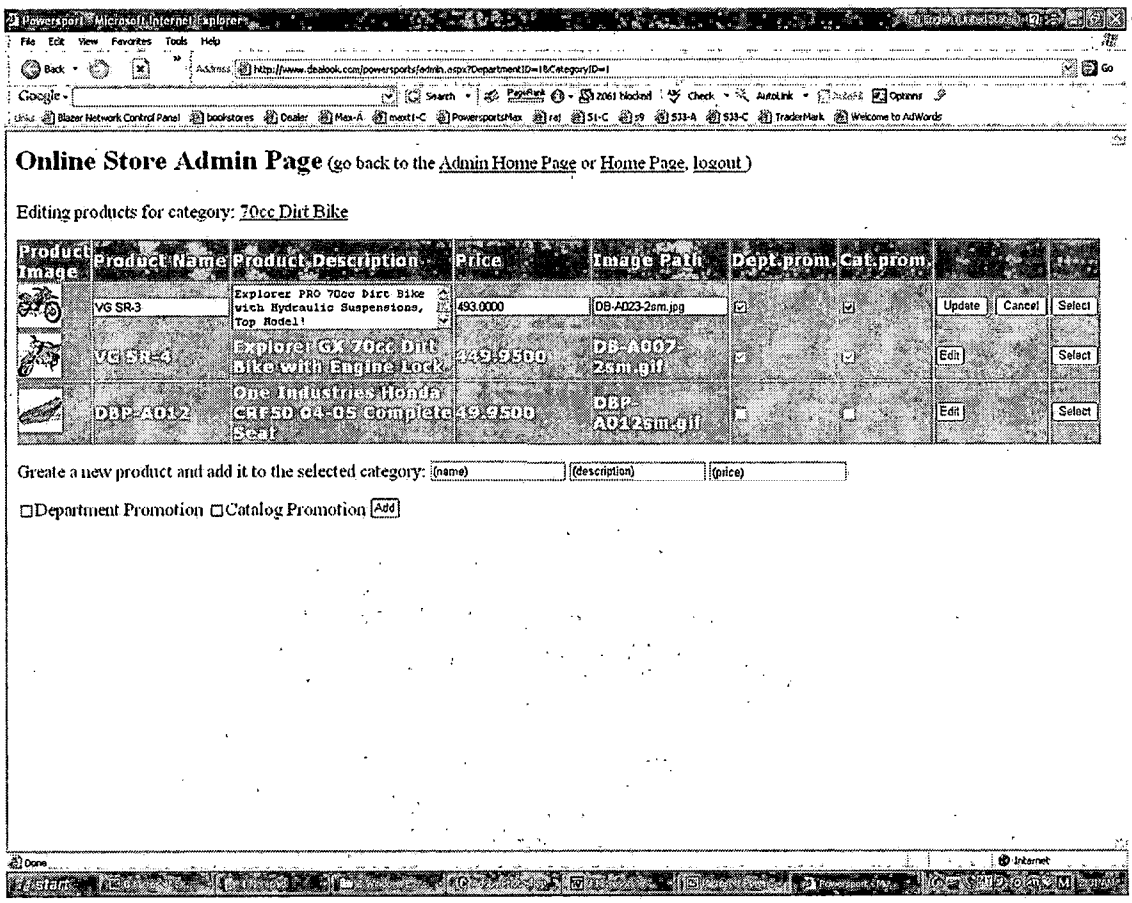

Figure 11. Product Admin Page

In this product details page, administrator can assign and move the product to different catalog, also can upload product images by browsing.

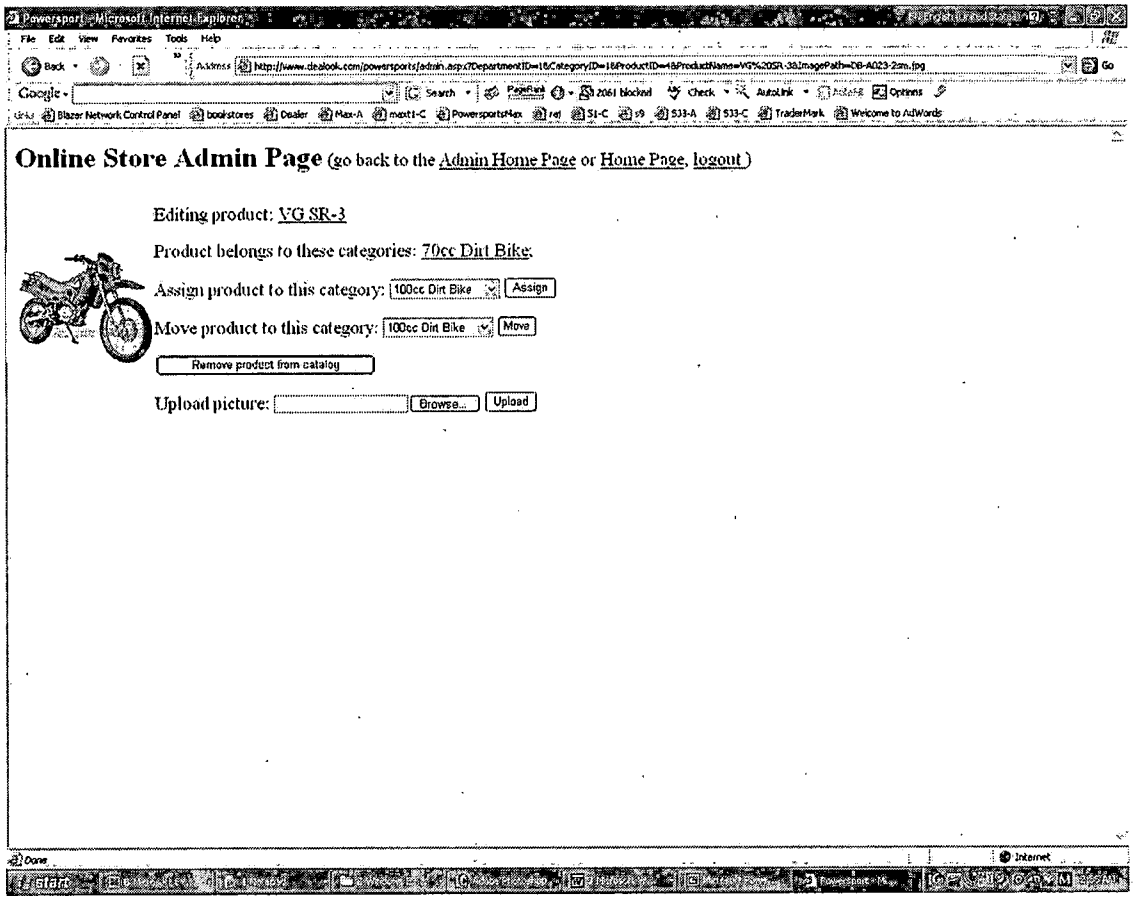

Figure 12. Product Detail Page

4.2.4 Customer Admin Page

The administrator can search customers and get a customer record by customer ID or customer name. Administrator can change customer details by clicking "Change Customer Detail" link, change customer address and change customer credit card information. Also can delete this customer record in this page.

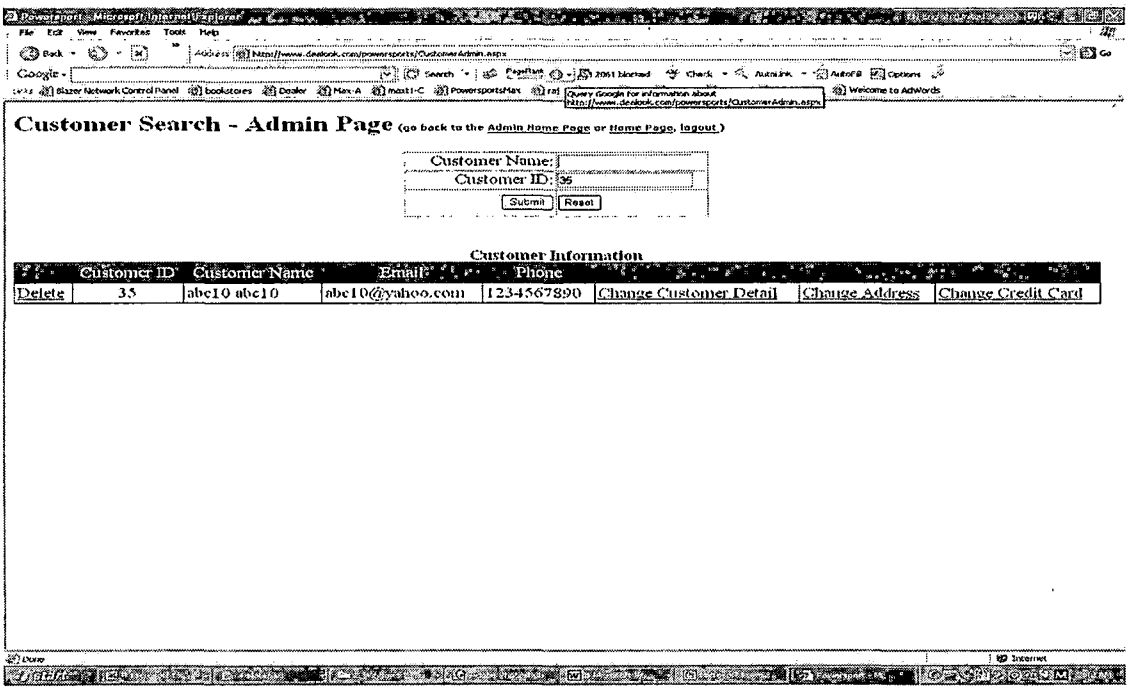

Figure 13. Customer Admin Page

Adminstorator can change customer password and phone number in this customer detail page.

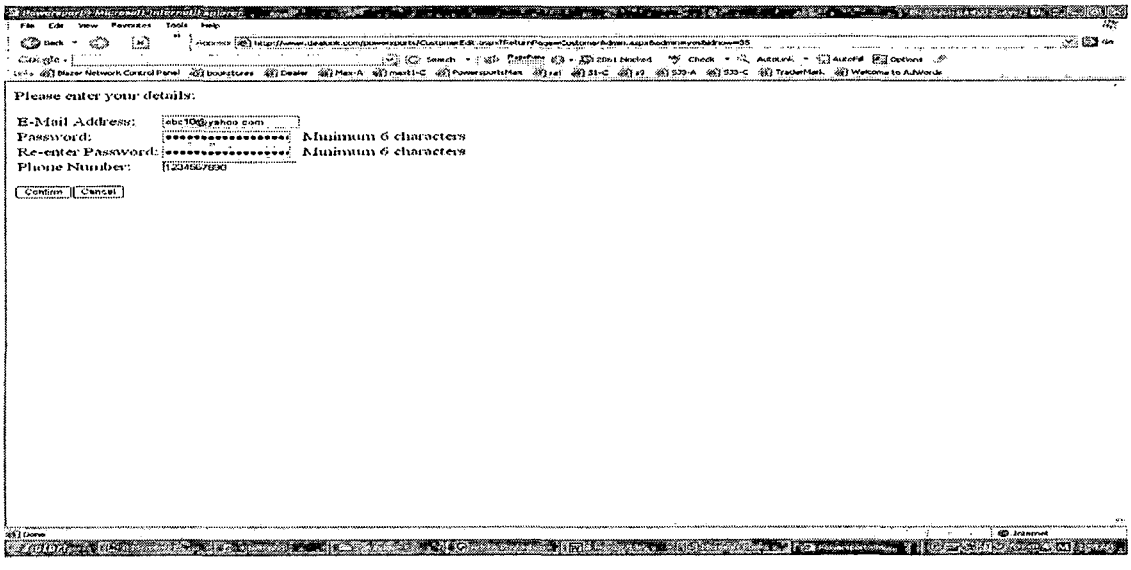

Figure 14. Change Customer Detail Page

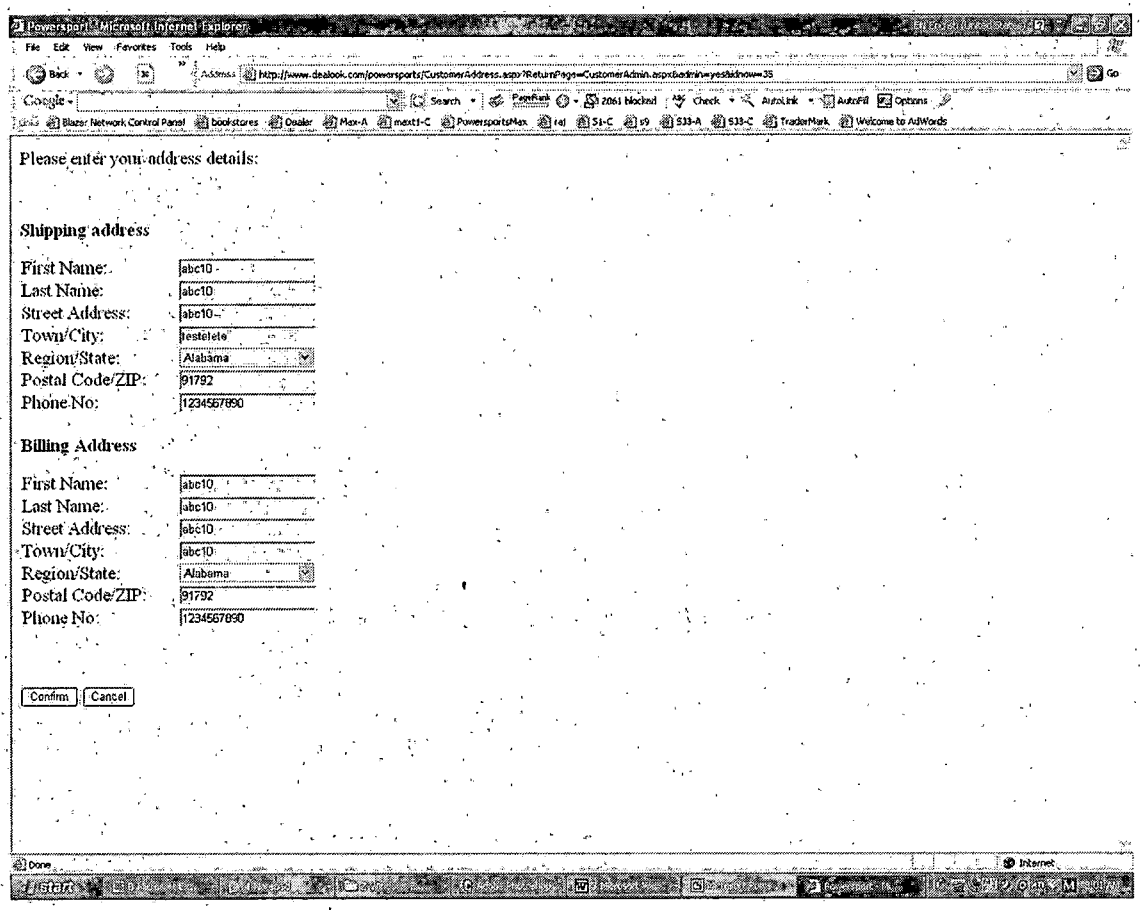

Figure 15. Change Customer Address Page

Adminstrator can change customer address in "Change customer address" page after clicking "Change Address" link.

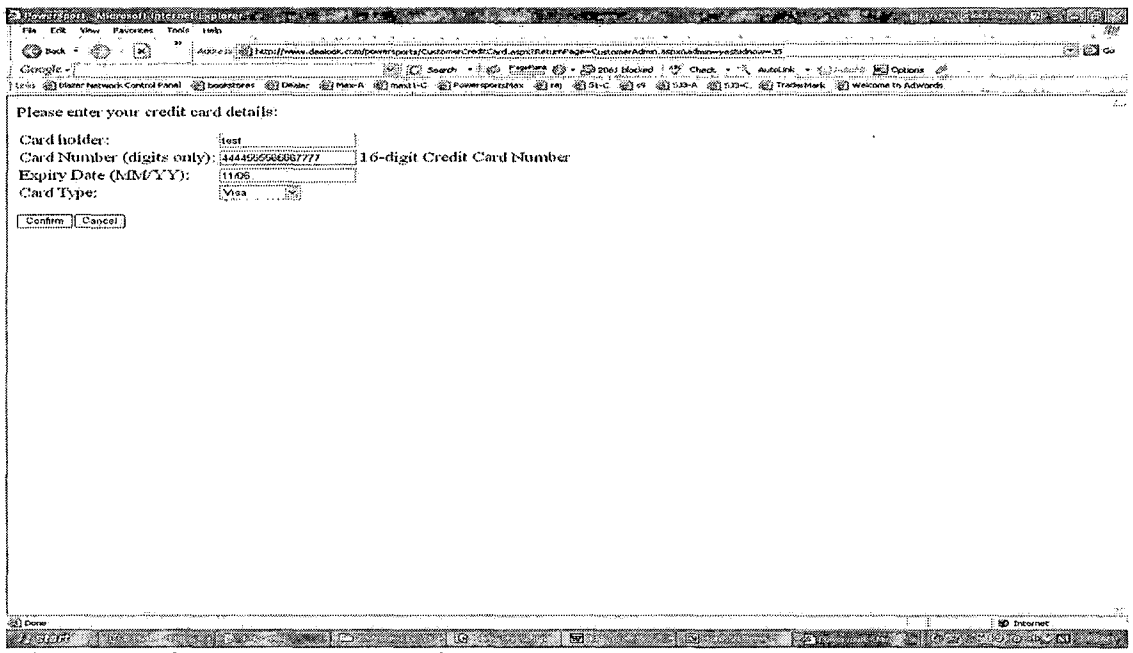

Figure 16. Change Credit Card Page

Adminstrator can change customer's credit card information in following page after clicking "Change Credit Card" link. We will get a Customers list page if Adminstrator deletes a customer record.

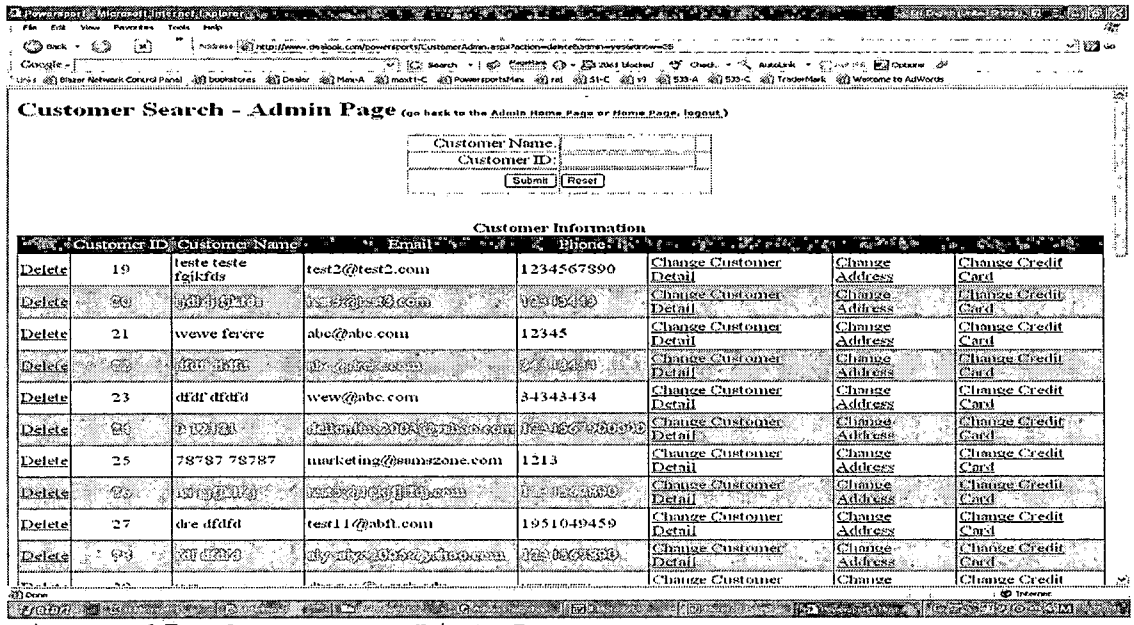

Figure 17. Customers List Page

4.2.5 Orders Admin Page

The Orders Admin page offers the capability to see the list of orders. When first loaded, it will offer us various ways to select orders, as shown in Figure 18.

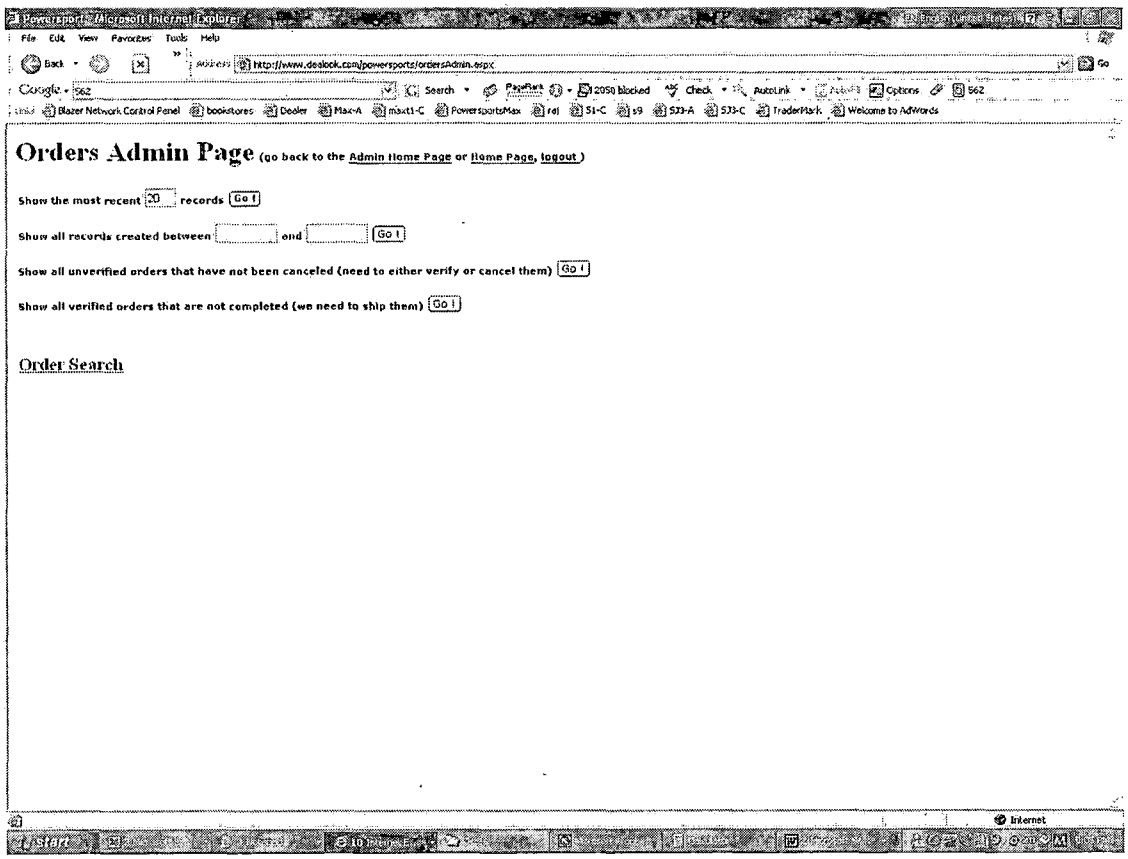

Figure 18. Orders Admin Page

Adminstrator can get a different recent orders list when input number in "Show the most recent orders" search function.

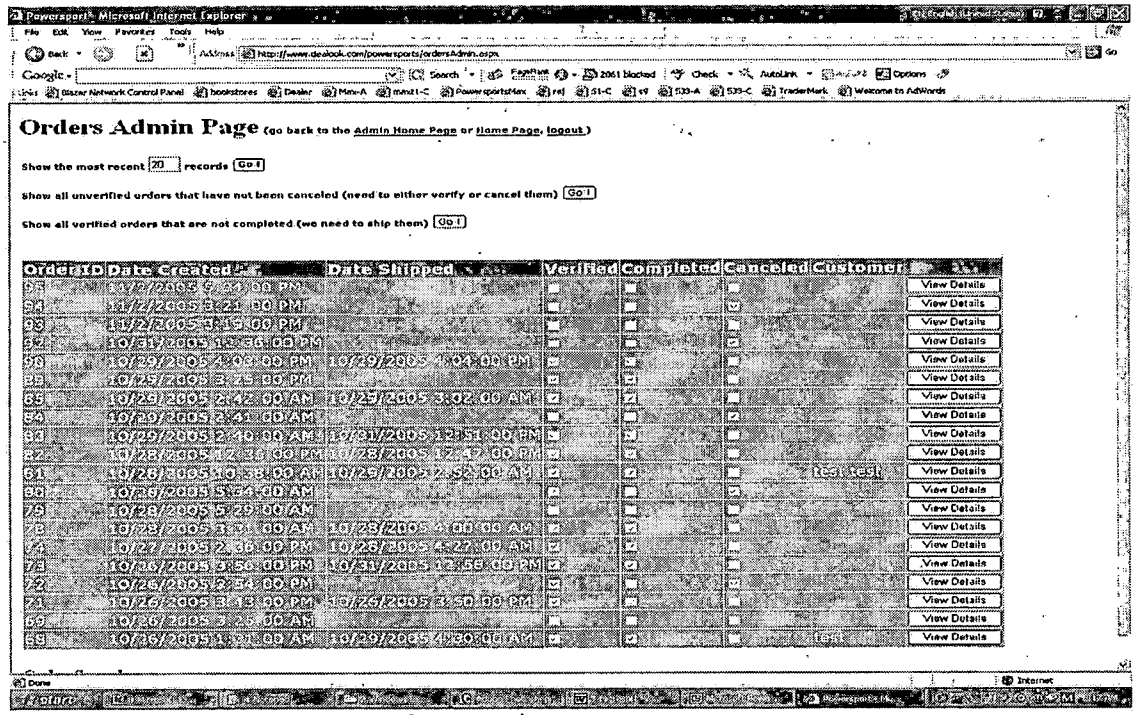

Figure 19. Recent Orders List Page

When adminstrator click "View Details" button next to the order, page will go to order details page. In this page, adminstrator can edit shipping, billing address and credit card information; also he can verify, complete or cancel the order.

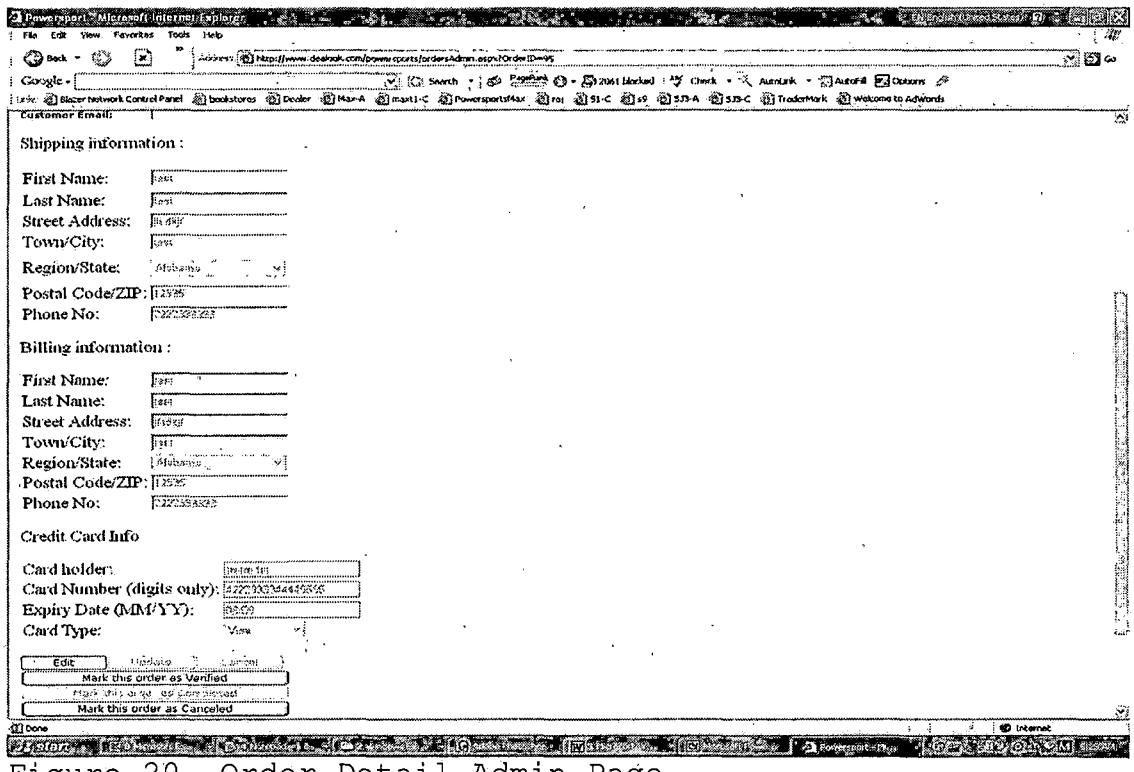

Figure 20. Order Detail Admin Page

4.2.6 Order Details Admin Page

In order search function administrator can get order records by searching customer name or order ID. When click "View Orders" in each order record, page will go to order details page.

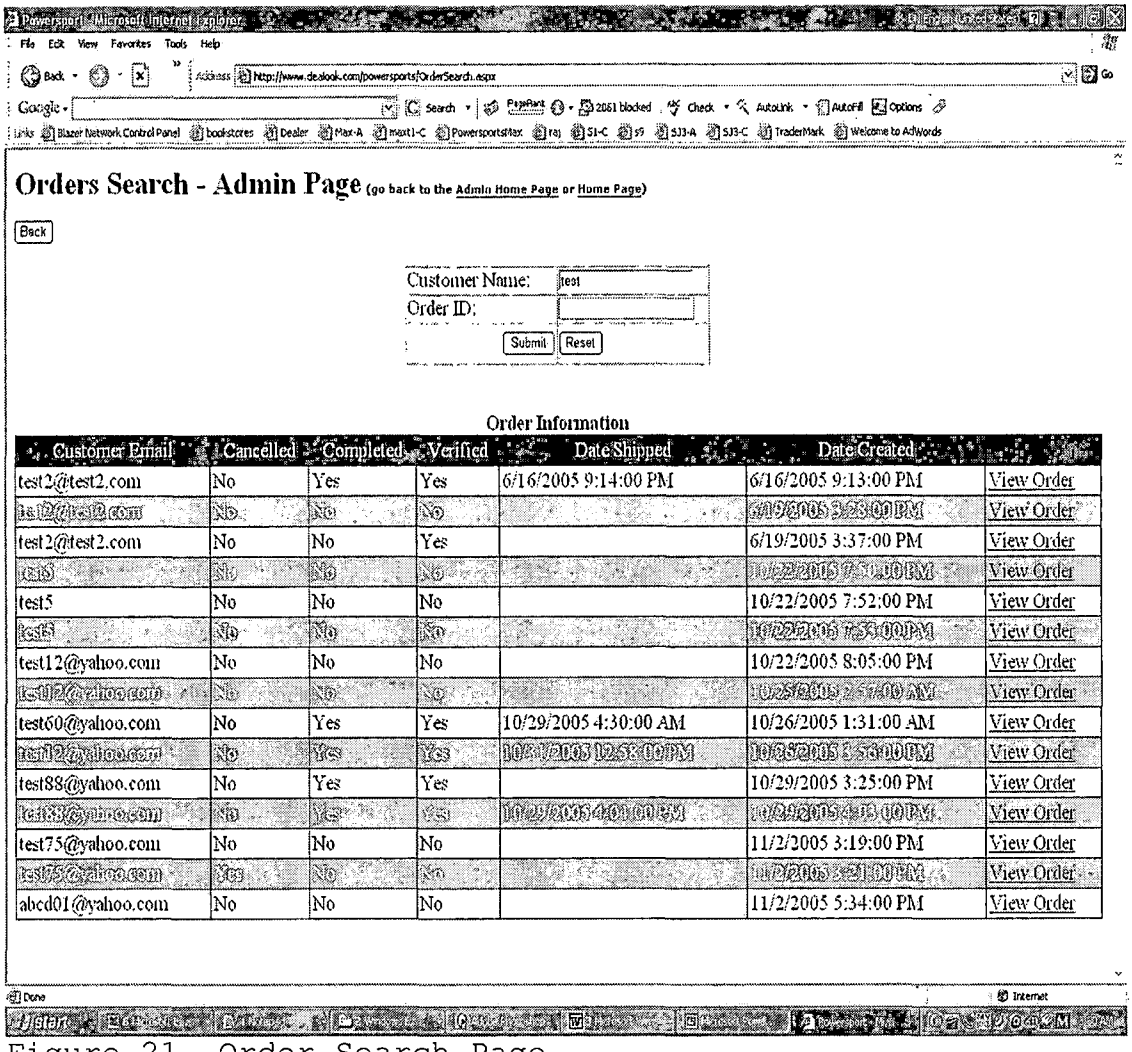

Figure 21. Order Search Page

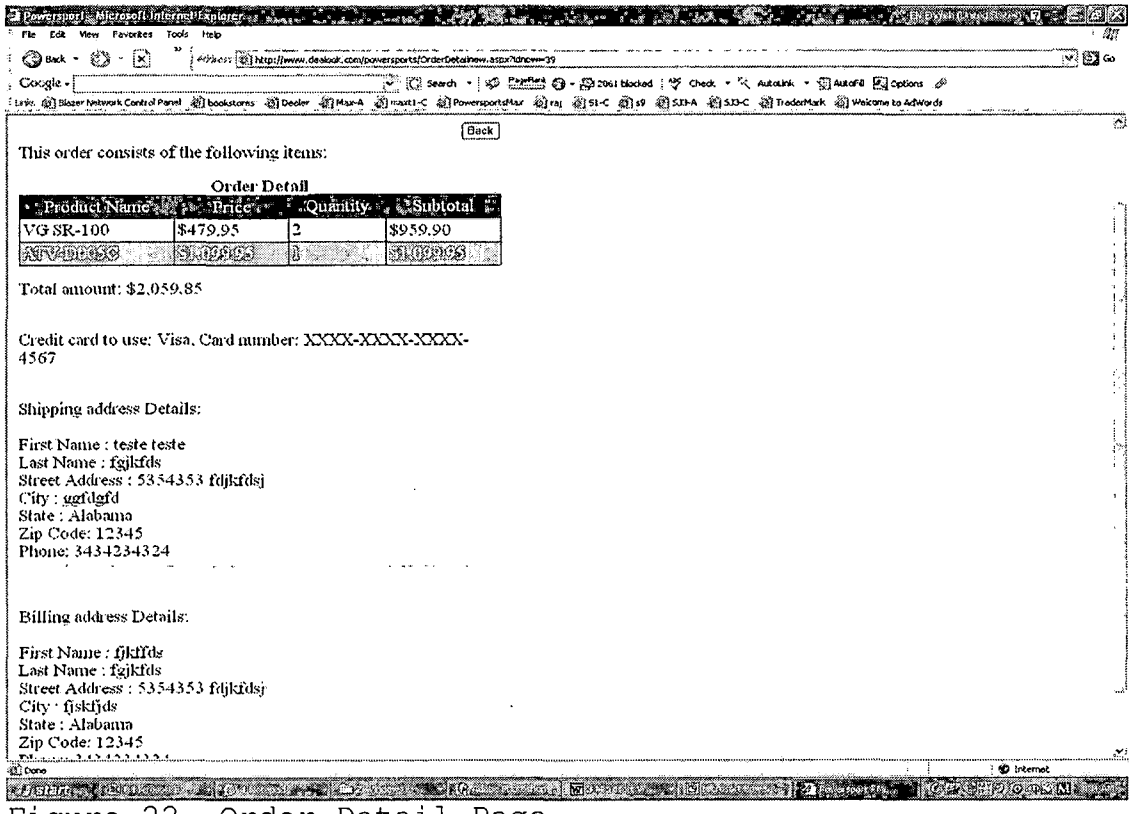

Figure 22. Order Detail Page

4.2.7 View Shopping Cart Page

Customers can update product quantity, remove product from shopping cart, and cancel the event.

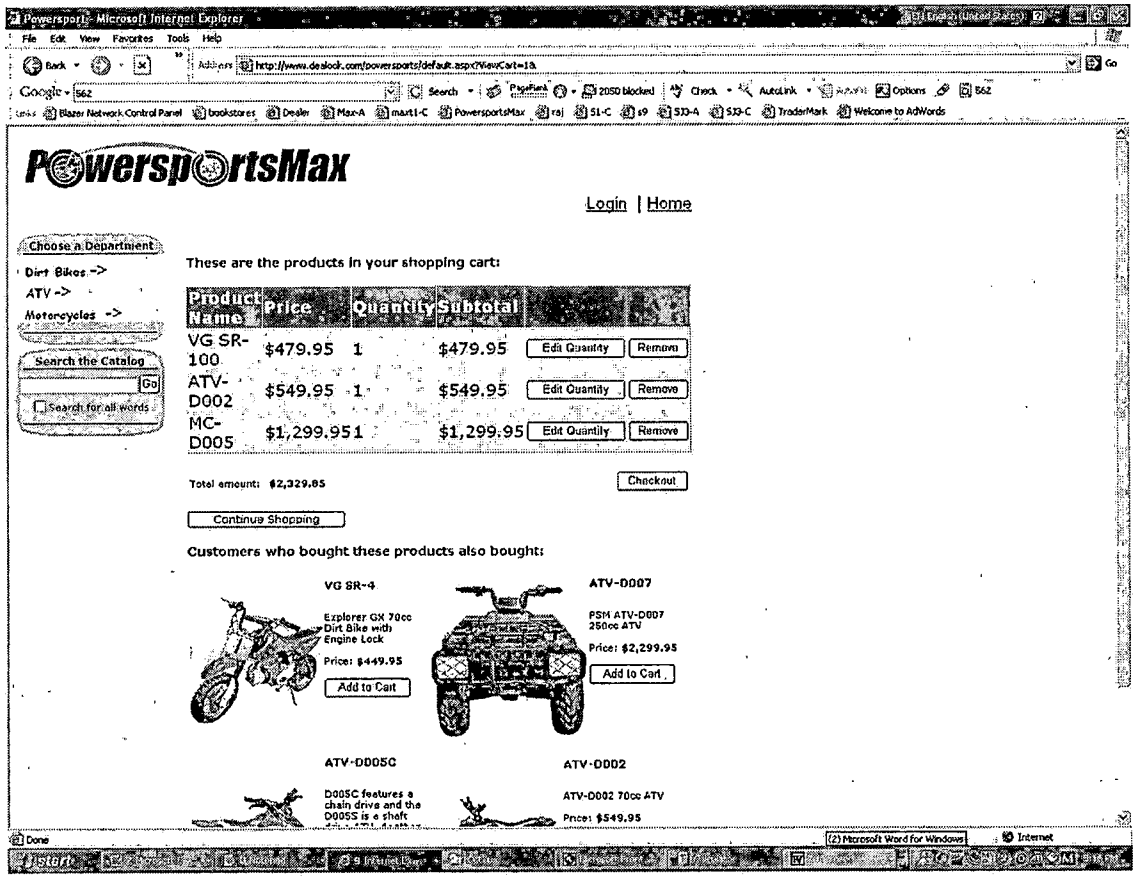

Figure 23. View Shopping Cart Page

4.2.8 Check Out Page

Customer can submit order by clicking "Place Order" Button.

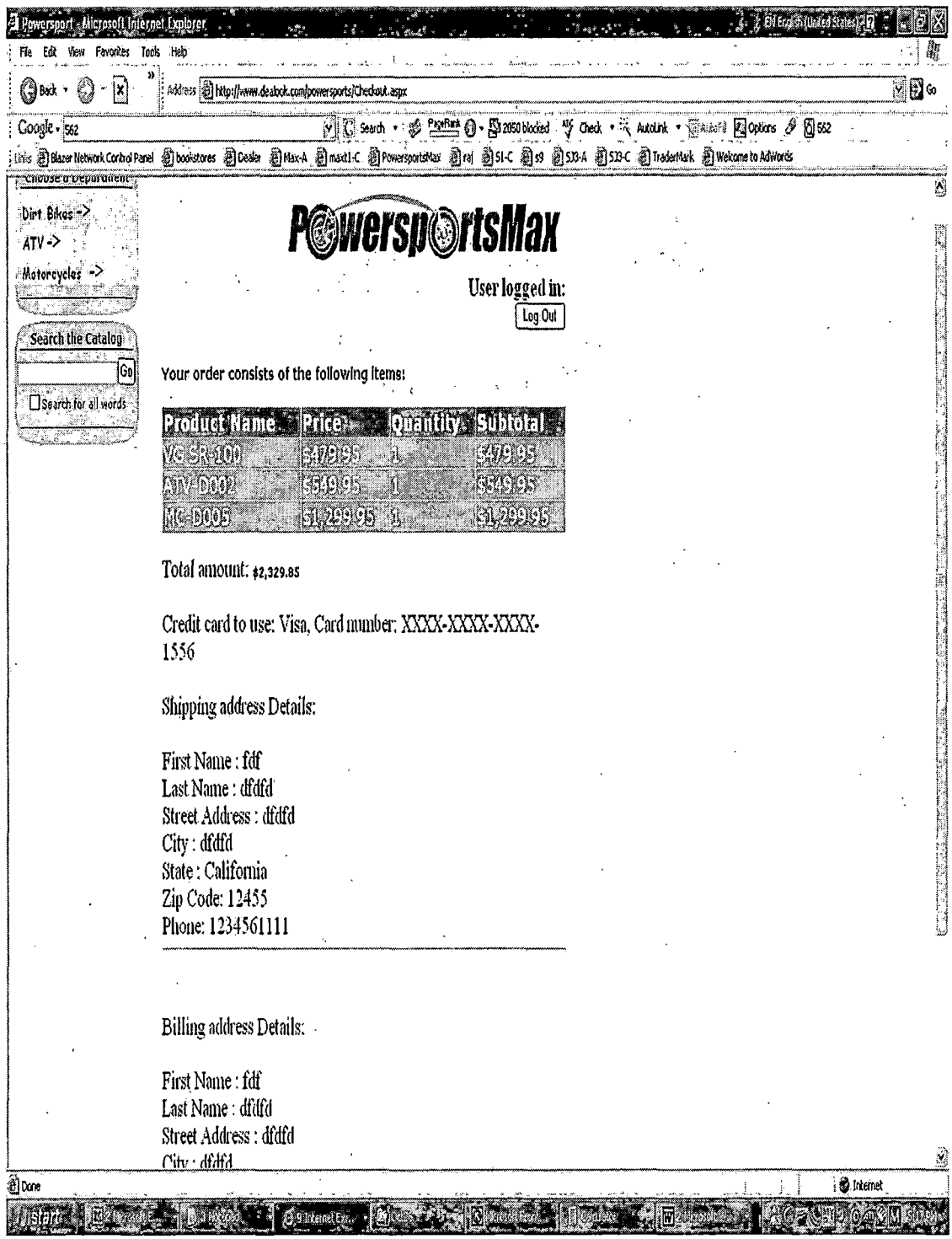

Figure 24. Customer Checkout Page

4.2.9 Customer Login Page

New Customer need register account information to log in. Also customer can update and edit the information after register.

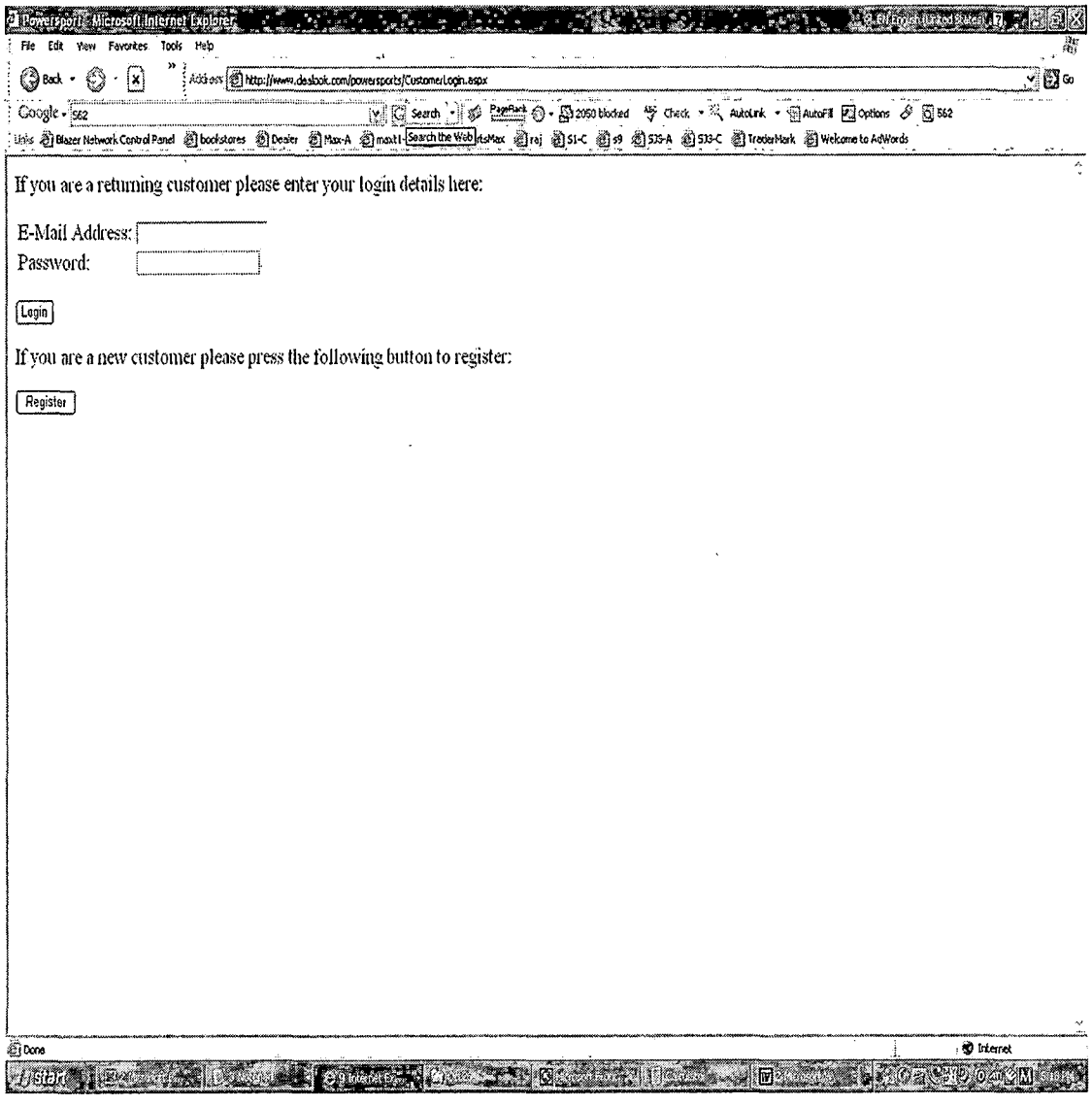

Figure 25. Customer login Page

4.3 Project Implementation

This section describes implementation of Catalog function, shopping cart, and order management.

4.3.1 Catalog Function Implementation

Department List

Page: DepartmentsList.ascx
Attributes: connectionString: string
Methods: Page_load: DepartmentsList

Figure 26. Department List Class

4.3.2 Shopping Cart Implementation

ShoppingCart class.

Page: ProductsList.ascx
Attributes: AddProduct: string UpdateProductQuantity: String RemoveProduct: String GetProductss: SqlDataReader GetTotalAmount: Decimal
Methods: Page_load: save DepartmentID from the query string to a variable

Figure 27. Shopping Cart Information Page Class

4.3.3 Order Management Implementation

Order Manager class.

Page: OrdersAdmin.aspx
Attributes: GetOrderInfo: String GetOrderDetails: String MarkOrderAsVerified: String MarkOrderAsCompleted: String MarkOrderAsCancelled: String
Methods: Page_load: Update information in all controls.

Figure 28. Order Manager Information Page Class

4.3.4 Customer Account Implementation

Customer Login Class

Page: CustomerLogin.aspx
Attributes: strQuery: string Super Class: clsMedasolution
Methods: Page_load: retrieve session user from home page function. Logout: logout the current user and clear all sessions.

Figure 29. Customer Account Information Page Class

Customer Edit page class.

Page: CustomerEdit.aspx
Attributes: strQuery: string clsdb: clsMedasolution
Methods: LoadName: retrieve customer name from database and LoadState: retrive state from database and load state name LoadMaritalStatus: retrieve maritalStatus from database and load maritalstatus status Databind: binding data into the labels

Figure 30. Customer Edit Information Page Class

CHAPTER FIVE

EVALUATION OF ASP .NET TECHNOLOGY

5.1 Introduction

In above section we already discussed how to design and implement an e-commerce website based the .NET technology. While ASP .NET technology is a very broad area and has a lot of components. In this chapter we will foucus on evaluation these two aspects: the cost of building an e-commerce site with ASP .NET technology; advantages and disadvantages of ASP .NET in building an e-commerce site.

5.2 Evaluation of ASP .NET Technology

In this section, will discuss the cost of builing e-commerce system with ASP .NET technology based on cost comparison tables [14].

As listed in the table below, the cost of building an ecommerce site based on .NET is low if the ecommerce site is hosted outside

Comparing to some open source solutions, like the popular PHP/My SQL/Apache platform, it costs more. The major difference is Windows 2003 Server and MS SQL.

Table 2. Cost Comparision If Hosting E-commerce Site Outside

	.NET	PHP
Software Price	Free	Free
Web server	Windows 2003 (Included in Hosting fee)	Apache (Free)
Platform	IIS Only	Any
Developer Tools	VB .NET ($50-$100)/Webmatrix (free)	Edit Tools, FTP ($50-100)
Technical Support	Yes	No
Database	$30/Month	$10/Month
Source Available	No	Yes

However, because many small to middle size businesses do not usually host their own server, instead they use outside hosting service.

Table 3. Cost Comparison If Hosting E-commerce Site On Its Own

	.NET	PHP
Software Price	Free	Free
Web server	Windows 2003 $2000	Apache (Free)
Platform	IIS Only	Any
Developer Tools	VB .NET ($50-$100)/Webmatrix (free)	Edit Tools, FTP ($50-100)
Technical Support	Yes	No
Database	SQL Server 2000 ($20000)	MYSQL (Free)
Source Available	No	Yes

The price difference between a Windows 2003/MS SQL server and Linux/Apache/MySQL is only $20 per month. This will be almost negligible in any enterprise project.

5.3 Advantages and Disadvantages of ASP .NET Technology

In this section, we will discuss the advantages and disadvantages of developing an e-commerce system with ASP .NET technology.

5.3.1 Advantages

While it may cost a bit more when initially building a .NET e-Commerce site, it has some advantages.

- Easy to develop

Regarding the server-side code, ASP .NET has the advantage of allowing this code to be written separately from the static part of the file. All the code that belongs to a certain page can be saved as code-behind file, which is an advantage over its major competitors: PHP, regular ASP, and JSP in which the HTML code and the server-side code were mixed in a single file. These mixed files are the subject to both designer's and programmers' work, and are hard to document, change and maintain.

- Excellent Microsoft Support

It is very easy to find all kinds of documents and tutorials from Microsoft website. There are also a lot training courses and tutorials available for .NET. For open source solution, a developer largely depends on open source community to get help. Sometimes it could be a frustrating like a developer ask a questions in a forum for long time, and no one answers the question. It is not so easy to find detailed and complete guide and documents as we do in Microsoft site.

- More Language Support

Another good thing about .NET is [15] that it has multi-language support. The server-side code can be written in the .NET language of your choice. The .NET Framework is supplied with C#, VB .NET, and Jscript.NET, but the whle infrastructure is designed to support additional languages. These languages are powerful and fully object oriented. What's nice about this is that developers can mix the code, can instantiate an object in C# from a class written in C++ or VB .NET. This improves developer's chances of finding a suitable pre-written class for developer's project on the Web.

- Good Development Environments

Microsoft has provided [16][17][18] a free development environment for ASP .NET called Web Matrix. It has a built-in Web server, database administration interface FTP integration, and more. Not only that, Microsoft has also released MSDE -- a free development edition of MS SQL server. It has precisely all the features of the full MS SQL server 2000, and any application developers write for MSDE will run fine on MS SQL Server.

Other than Web Matrix, Visual Studio .NET [17] is an excellent and complete visual editor, which make coding much, much easier and even seasoned developers more productive. It can highlight syntax, let developers know when the wrong stuff is commented, do command completion, and just plain help developers organize better. Visual Studio .NET also can be used as an interface to the database to create table and store procedures, implement tables replationships, and so on.

5.3.2 Disadvantages

ASP .NET is a powerful language and fully object oriented, but it has some disadvantages too.

- The learning curve is stiff

ASP .NET is written using "real" OO (Object Oriented) programming languages choice. Comparing to PHP, .NET languages like C++, VB .NET or C# -- are also harder to learn and master, programmers without much coding backgrounds may take more time to master ASP .NET.

- The initial set-up cost is higher

If a business plans to host its ecommerce site on its own. The cost is much higher than PHP/My SQL/Apache Solution.

Window 2003 Server Enterprise Version costs $2000, and SQL server 2000 for a Single CPU cost $20000, they are pretty expensive for small businesses.

- Cross-platform applicability is not strong

Comparing to PHP or JSP, .NET technology run on less platforms. .NET works great with Microsoft windows platforms. ASP .NET does not have the same compabitility with Apache server as PHP has.

CHAPTER SIX

CONCLUSTION

6.1 Conclusion

In this project, we first discuss procedures and major components of an e-commerce shopping cart. In next step, we build up a platform for an ASP .NET based shopping cart, which includes, IIS or Windows 2003 server, SQL database and Visual studio development tools. Under this platform, we use ASP .NET to develop major modules of an e-commerce shopping cart. We first developed front-end catalog to display products for customers. The look and feel of catalog is easily to modify as we use "User Control" to control them. We also developed shopping cart to take orders and integrated it with Paypal gateway, which enable our shopping cart to take credit card online. We created back-end section so that administrator can manage orders and customers can edit shopping cart. With front-end, back-end and the integration with Paypal gateway service, our e-commerce application covers major parts of a shopping cart system and can be used in real world.

Finally, we evaluate ASP .NET technology based on our experience from this project. We evaluate it from these aspects:

1) The cost analysis to develop and implement ASP .NET based ecommerce system.

2) The effectiveness of developing an ecommerce system with ASP .NET.

In this project, we discussed and displayed how to use ASP .NET technology to develop an ecommerce system. We concluded that it is cost effective to build an e-commerce shopping cart based on ASP .NET. It is not free but very affordable. With built-in techniques in .NET technology, it is fast to build a robust and reliable shopping cart system with ASP .NET.

6.2 Future Direction

Even though we have achieved the objectives and goals that we aimed in this project, there are still some points needed to be addressed for future directions due to it potential practical usefulness. Comparing with pratical shopping cart, this project is lack of actual payment method which can be developped in future.

In the real world shopping cart system, it should give customers optipons to pay by credit cards and process

payment throught payment gateway. We can develop a payment gateway module so that the shopping cart can work with popuplar payment gateways like Authorize .NET, Linkpoint, Paypal, etc.

APPENDIX A

ASP .NET FILE

admin.aspx

```
<%@ Register TagPrefix="uc1" TagName="DepartmentsAdmin"
Src="AdminUserControls/DepartmentsAdmin.ascx" %>
<%@ Register TagPrefix="uc1" TagName="CategoriesAdmin"
Src="AdminUserControls/CategoriesAdmin.ascx" %>
<%@ Register TagPrefix="uc1" TagName="ProductsAdmin"
Src="AdminUserControls/ProductsAdmin.ascx" %>
<%@ Register TagPrefix="uc1" TagName="Logout" Src="AdminUserControls/Logout.ascx" %>
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="admin.aspx.vb"
Inherits="JokePoint.admin"%>
<%@ Register TagPrefix="uc1" TagName="ProductDetailsAdmin"
Src="AdminUserControls/ProductDetailsAdmin.ascx" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
        <HEAD>
                <title>admin</title>
                <meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.1">
                <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
                <meta name="vs_defaultClientScript" content="JavaScript">
                <meta name="vs_targetSchema"
content="http://schemas.microsoft.com/intellisense/ie5">
                <link href="JokePoint.css" type="text/css" rel="stylesheet">
        </HEAD>
        <body>
                <form id="Form1" method="post" runat="server">
                <strong><font size="5"> Admin Page</font></strong>
                        <uc1:Logout id="Logout1" runat="server"></uc1:Logout>
                        <br>
                        <br>
                <table border="0" cellpadding="0">
                        <tr>
                         <td>
                        <uc1:DepartmentsAdmin id="DepartmentsAdmin1" runat="server">
        </uc1:DepartmentsAdmin>
                         </td>
                        </tr>
                        <tr>
                        <td>
                        <uc1:CategoriesAdmin id="CategoriesAdmin1" runat="server">
        </uc1:CategoriesAdmin>
                        </td>
                        </tr>
                        <tr>
                         <td>
                         <uc1:ProductsAdmin id="ProductsAdmin1" runat="server">
        </uc1:ProductsAdmin>
                         </td>
                        </tr>
                        <tr>
                         <td>
```

```
                    <uc1:ProductDetailsAdmin id="ProductDetailsAdmin1" runat="server">
        </uc1:ProductDetailsAdmin>
                    </td>
                    </tr>
                    </table>
                    </form>
            </body>
        </HTML>
```

default.aspx

```
        <%@ Register TagPrefix="uc1" TagName="CategoriesList"
Src="UserControls/CategoriesList.ascx" %>
        <%@ Register TagPrefix="uc1" TagName="DepartmentsList"
Src="UserControls/DepartmentsList.ascx" %>
        <%@ Page Language="vb" AutoEventWireup="false" Codebehind="default.aspx.vb"
Inherits="JokePoint._default"%>
        <%@ Register TagPrefix="uc1" TagName="SearchBox" Src="UserControls/SearchBox.ascx"
%>
        <%@ Register TagPrefix="uc1" TagName="Header" Src="UserControls/Header.ascx" %>
        <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
        <HTML>
                <HEAD>
                    <title>Gradate Project</title>
                    <meta name="GENERATOR" content="Microsoft Visual Studio .NET
7.1">
                    <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
                    <meta name="vs_defaultClientScript" content="JavaScript">
                    <meta name="vs_targetSchema"
content="http://schemas.microsoft.com/intellisense/ie5">
                    <link href="JokePoint.css" type="text/css" rel="stylesheet">
                    <script language="JavaScript">
          <!--
          var PayPalWindow = null;

          function OpenPayPalWindow(url)
          {
           if ((!PayPalWindow) || PayPalWindow.closed)
            // If the PayPal window doesn't exist, we open it
            PayPalWindow = window.open(url,"cart","height=300, width=500");
           else
           {
            // If the PayPal window exists, we make it show
            PayPalWindow.location.href=url;
            PayPalWindow.focus();
           }
          }

          // -->
                    </script>
```

```
		</HEAD>
		<body>
			<form id="Form1" method="post" runat="server">
			<table height="100%" cellSpacing="0" cellPadding="0" width="770"
border="0">
			<tr>
			<td width="190" height="100%">
			<table height="100%" width="190" cellspacing="0" cellpadding="0">
			<tr>
			<td vAlign="top" height="100%">
			<uc1:DepartmentsList id="DepartmentsList1"
runat="server"></uc1:DepartmentsList>
			<uc1:CategoriesList id="CategoriesList1"
runat="server"></uc1:CategoriesList>
			<uc1:SearchBox id="SearchBox1" runat="server"></uc1:SearchBox>
			<br>
	         
			<asp:Button id="viewCartButton" runat="server" Text="View
Cart"></asp:Button>
			</td>
			</tr>
			</table>
			</td>
			<td vAlign="top" width="550"><br>
			<table>
			<tr>
			<td>
			<uc1:Header id="Header1" runat="server"></uc1:Header></td>
			</tr>
			<tr>
			<td id="pageContentsCell" runat="server">
			</td>
			</tr>
			</table>
			<P> </P>
			</td>
			</tr>
			</table>
			</form>
		</body>
	</HTML>
```

login.aspx

```
	<%@ Page Language="vb" AutoEventWireup="false" Codebehind="login.aspx.vb"
Inherits="JokePoint.login"%>
	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
	<HTML>
		<HEAD>
			<title>login</title>
```

```
                    <meta name="GENERATOR" content="Microsoft Visual Studio .NET
7.1">
                    <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
                    <meta name="vs_defaultClientScript" content="JavaScript">
                    <meta name="vs_targetSchema"
        content="http://schemas.microsoft.com/intellisense/ie5">
                </HEAD>
                <body>
                    <form id="Form1" method="post" runat="server">
                        <P>
                        <asp:Label id="loginMessageLabel"
runat="server"></asp:Label></P>
                        <P>
                        <asp:Label id="userNameLabel" runat="server">User
Name:</asp:Label>
                        <asp:TextBox id="userNameTextBox"
runat="server"></asp:TextBox><BR>
                        <asp:Label id="passwordLabel"
runat="server">Password:</asp:Label>
                        <asp:TextBox id="passwordTextBox" runat="server"
TextMode="Password"></asp:TextBox></P>
                        <P>
                        <asp:CheckBox id="persistCheckBox" runat="server"
Text="Persist Security Info"></asp:CheckBox></P>
                        <P>
                        <asp:Button id="loginButton" runat="server"
Text="Login"></asp:Button></P>
                    </form>
                </body>
        </HTML>
```

CustomerLogin.aspx

```
        <%@ Page Language="vb" AutoEventWireup="false" Codebehind="CustomerLogin.aspx.vb"
Inherits="JokePoint.CustomerLogin" %>
        <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
        <HTML>
          <HEAD>
            <title>Customer Login</title>
            <meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.1">
            <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
            <meta name="vs_defaultClientScript" content="JavaScript">
            <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
          </HEAD>
          <body MS_POSITIONING="GridLayout">
            <form id="Form1" method="post" runat="server">
              <p>If you are a returning customer please enter your login details here:</p>
              <P>
                <table>
                  <tr>
```

```
            <td><asp:label id="Label1" runat="server" text="E-Mail Address:" /></td>
            <td><asp:textbox id="txtEmail" runat="server" /></td>
          </tr>
          <tr>
            <td><asp:label id="Label2" runat="server" text="Password:" /></td>
            <td><asp:textbox id="txtPassword" runat="server" TextMode="Password" /></td>
          </tr>
        </table>
      </P>
      <P><asp:button id="btnLogin" runat="server" Text="Login"></asp:button></P>
      <P><asp:label id="lblLoginMsg" runat="server"></asp:label></P>
      <P>If you are a new customer please press the following button to register</P>
      <P><asp:button id="btnRegister" runat="server" Text="Register"></asp:button></P>
    </form>
  </body>
</HTML>
```

CustomerAddress.aspx

```
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="CustomerAddress.aspx.vb"
Inherits="JokePoint.CustomerAddress" %>
    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
    <HTML>
        <HEAD>
            <title>Address Details</title>
            <meta content="Microsoft Visual Studio .NET 7.0"
name="GENERATOR">
            <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
            <meta content="JavaScript" name="vs_defaultClientScript">
            <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
        </HEAD>
        <body>
            <form id="Form1" method="post" runat="server">
                <p>Please enter your address details:</p>
        <p><table>
            <tr>
                <td><asp:label id="Label3" runat="server">Address
1:</asp:label></td>
                <td><asp:textbox id="txtAddress1"
runat="server"></asp:textbox></td>
                <td><asp:RequiredFieldValidator ID="validateAddress1"
Runat="server" ControlToValidate="txtAddress1" ErrorMessage="You must enter an address.">
    </asp:RequiredFieldValidator></td>
            </tr>
            <tr>
            <td><asp:label id="Label1" runat="server">Address 2:</asp:label></td>
            <td><asp:textbox id="txtAddress2" runat="server"></asp:textbox></td>
            <td></td>
            </tr>
```

```
                        <tr>
                        <td><asp:label id="Label2" runat="server">Town/City:</asp:label></td>
                        <td><asp:textbox id="txtCity" runat="server"></asp:textbox></td>
                        <td><asp:RequiredFieldValidator ID="validateCity" Runat="server"
ControlToValidate="txtCity" ErrorMessage="You must enter a
town/city."></asp:RequiredFieldValidator></td>
                        </tr>
                        <tr>
                        <td><asp:label id="Label4" runat="server">Region/State:</asp:label></td>
                        <td><asp:textbox id="txtRegion" runat="server"></asp:textbox></td>
                        <td><asp:RequiredFieldValidator ID="validateRegion" Runat="server"
ControlToValidate="txtRegion" ErrorMessage="You must enter a
region/state."></asp:RequiredFieldValidator></td>
                        </tr>
                        <tr>
                        <td><asp:label id="Label5" runat="server">Postal
Code/ZIP:</asp:label></td>
                        <td><asp:textbox id="txtPostalCode" runat="server"></asp:textbox></td>
                        <td><asp:RequiredFieldValidator ID="validatePostalCode" Runat="server"
ControlToValidate="txtPostalCode" ErrorMessage="You must enter a postal
code/ZIP."></asp:RequiredFieldValidator></td>
                        </tr>
                        <tr>
                        <td><asp:label id="Label6" runat="server">Country:</asp:label></td>
                        <td><asp:textbox id="txtCountry" runat="server"></asp:textbox></td>
                        <td><asp:RequiredFieldValidator ID="validateCountry" Runat="server"
ControlToValidate="txtCountry" ErrorMessage="You must enter a
country."></asp:RequiredFieldValidator></td>
                        </tr>
                </table>
                </p>
        <P><asp:button id="btnConfirm" runat="server" Text="Confirm"></asp:button><asp:button
id="cancelButton" runat="server" Text="Cancel"></asp:button></P>
        <P><asp:label id="lblMsg" runat="server"></asp:label></P>
                        </form>
                </body>
        </HTML>
```

CustomerCreditCard.aspx

```
        <%@ Page Language="vb" AutoEventWireup="false"
Codebehind="CustomerCreditCard.aspx.vb" Inherits="JokePoint.CustomerCreditCard" %>
        <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
        <HTML>
                <HEAD>
                        <title>Credit Card Details</title>
                        <meta content="Microsoft Visual Studio .NET 7.0"
name="GENERATOR">
                        <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
                        <meta content="JavaScript" name="vs_defaultClientScript">
```

```
                    <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
                </HEAD>
        <body>
        <form id="Form1" method="post" runat="server">
                <p>Please enter your credit card details:</p>
                <P><table>
                        <tr>
                        <td><asp:label id="Label3" runat="server">Card holder:</asp:label></td>
                        <td><asp:textbox id="txtCardHolder" runat="server"></asp:textbox></td>
                        <td><asp:RequiredFieldValidator ID="validateCardHolder" Runat="server"
ControlToValidate="txtCardHolder" ErrorMessage="You must enter a card
holder."></asp:RequiredFieldValidator></td>
                        </tr>
                        <tr>
                        <td><asp:label id="Label1" runat="server">Card Number (digits
only):</asp:label></td>
                        <td><asp:textbox id="txtCardNumber" runat="server"></asp:textbox></td>
                        <td><asp:RequiredFieldValidator ID="validateCardNumber"
Runat="server" ControlToValidate="txtCardNumber" ErrorMessage="You must enter a card
number."></asp:RequiredFieldValidator></td>
                        </tr>
                        <tr>
                        <td><asp:label id="Label2" runat="server">Expiry Date
(MM/YY):</asp:label></td>
                        <td><asp:textbox id="txtExpDate" runat="server"></asp:textbox></td>
                        <td><asp:RequiredFieldValidator ID="validateExpDate" Runat="server"
ControlToValidate="txtExpDate" ErrorMessage="You must enter an expiry
date."></asp:RequiredFieldValidator></td>
                        </tr>
                        <tr>
                        <td><asp:label id="Label4" runat="server">Issue Date (MM/YY if
applicable):</asp:label></td>
                        <td><asp:textbox id="txtIssueDate" runat="server"></asp:textbox></td>
                        <td></td>
                        </tr>
                        <tr>
                        <td><asp:label id="Label5" runat="server">Issue Number (if
applicable):</asp:label></td>
                        <td><asp:textbox id="txtIssueNumber" runat="server"></asp:textbox></td>
                                                <td></td>
                        </tr>
                        <tr>
                        <td><asp:label id="Label6" runat="server">Card Type:</asp:label></td>
                        <td><asp:dropdownlist id="txtCardType" runat="server">
                <asp:ListItem Value="Visa">Visa</asp:ListItem>
                <asp:ListItem Value="Mastercard">Mastercard</asp:ListItem>
                <asp:ListItem Value="Switch">Switch</asp:ListItem>
                <asp:ListItem Value="Solo">Solo</asp:ListItem>
                <asp:ListItem Value="American Express">American Express</asp:ListItem>
                </asp:dropdownlist></td>
```

```
                <td><asp:RequiredFieldValidator ID="validateCardType" Runat="server"
ControlToValidate="txtCardType" ErrorMessage="You must enter a card
type."></asp:RequiredFieldValidator>
                </td>
                </tr>
        </table>
                </P>
        <P><asp:button id="btnConfirm" runat="server" Text="Confirm"></asp:button><asp:button
id="cancelButton" runat="server" Text="Cancel"></asp:button></P>
        <P></P>
        </form>
                </body>
        </HTML>
```

CustomerEdit.aspx

```
        <%@ Page Language="vb" AutoEventWireup="false" Codebehind="CustomerEdit.aspx.vb"
Inherits="JokePoint.CustomerEdit" %>
        <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
        <HTML>
                <HEAD>
                        <title>Customer Details</title>
                        <meta content="Microsoft Visual Studio .NET 7.0"
name="GENERATOR">
                        <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
                        <meta content="JavaScript" name="vs_defaultClientScript">
                        <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
                </HEAD>
        <body>
                <form id="Form1" method="post" runat="server">
                <p>Please enter your details:</p>
                <table>
                  <tr>
                  <td><asp:label id="Label3" runat="server">Username:</asp:label></td>
                  <td><asp:textbox id="txtUserName" runat="server"></asp:textbox></td>
                  <td><asp:requiredfieldvalidator id="validateUserName" ErrorMessage="You must
enter a user name." ControlToValidate="txtUserName"
Runat="server"></asp:requiredfieldvalidator></td>
                </tr>
                <tr>
                        <td><asp:label id="Label1" runat="server">E-Mail
Address:</asp:label></td>
                        <td><asp:textbox id="txtEmail" runat="server"></asp:textbox></td>
                        <td><asp:requiredfieldvalidator id="validateEmail" ErrorMessage="You
must enter an e-mail address." ControlToValidate="txtEmail"
Runat="server"></asp:requiredfieldvalidator></td>
                </tr>
                <tr>
                        <td><asp:label id="Label2" runat="server">Password:</asp:label></td>
```

```
                    <td><asp:textbox id="txtPassword" runat="server"
TextMode="Password"></asp:textbox></td>
                    <td><asp:requiredfieldvalidator id="validatePassword" ErrorMessage="You
must enter a password." ControlToValidate="txtPassword"
Runat="server"></asp:requiredfieldvalidator></td>
                </tr>
                <tr>
                    <td><asp:label id="Label4" runat="server">Re-enter
Password:</asp:label></td>
                    <td><asp:textbox id="txtPasswordConfirm" runat="server"
TextMode="Password">
        </asp:textbox></td>
                    <td><asp:requiredfieldvalidator id="validatePasswordReEntry"
ErrorMessage="You must re-enter your password." ControlToValidate="txtPasswordConfirm"
Runat="server">
        </asp:requiredfieldvalidator>
        <asp:comparevalidator id="validatePasswordMatch" ErrorMessage="You must re-enter the
same password." ControlToValidate="txtPassword" Runat="server" Operator="Equal"
ControlToCompare="txtPasswordConfirm">
        </asp:comparevalidator></td>
                </tr>
                <tr>
                    <td><asp:label id="Label5" runat="server">Phone
Number:</asp:label></td>
                    <td><asp:textbox id="txtPhone" runat="server"></asp:textbox></td>
                    <td><asp:requiredfieldvalidator id="validatePhone" ErrorMessage="You
must enter a phone number." ControlToValidate="txtPhone"
Runat="server"></asp:requiredfieldvalidator></td>
                                        </tr>
        </table>
                <P></P>
                <P><asp:button id="btnConfirm" runat="server"
Text="Confirm"></asp:button><asp:button id="cancelButton" runat="server"
Text="Cancel"></asp:button></P>
                <P><asp:label id="lblMsg" runat="server"></asp:label></P>
                </form>
         </body>
        </HTML>
```

CustomerNew.aspx

```
        <%@ Page Language="vb" AutoEventWireup="false" Codebehind="CustomerNew.aspx.vb"
Inherits="JokePoint.CustomerNew" %>
        <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
        <HTML>
          <HEAD>
            <title>New Customer</title>
            <meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.1">
            <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
            <meta name="vs_defaultClientScript" content="JavaScript">
```

```
    <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
   </HEAD>
   <body MS_POSITIONING="GridLayout">
    <form id="Form1" method="post" runat="server">
     <p>Please enter your details:</p>
     <P>
      <table>
       <tr>
        <td><asp:label id="Label3" runat="server">Username:</asp:label></td>
        <td><asp:textbox id="txtUserName" runat="server"></asp:textbox></td>
        <td><asp:requiredfieldvalidator id="validateUserName" ErrorMessage="You must
enter a user name." ControlToValidate="txtUserName"
          Runat="server"></asp:requiredfieldvalidator></td>
       </tr>
       <tr>
        <td><asp:label id="Label1" runat="server">E-Mail Address:</asp:label></td>
        <td><asp:textbox id="txtEmail" runat="server"></asp:textbox></td>
        <td><asp:requiredfieldvalidator id="validateEmail" ErrorMessage="You must enter an
e-mail address." ControlToValidate="txtEmail"
          Runat="server"></asp:requiredfieldvalidator></td>
       </tr>
       <tr>
        <td><asp:label id="Label2" runat="server">Password:</asp:label></td>
        <td><asp:textbox id="txtPassword" runat="server"
TextMode="Password"></asp:textbox></td>
        <td><asp:requiredfieldvalidator id="validatePassword" ErrorMessage="You must
enter a password." ControlToValidate="txtPassword"
          Runat="server"></asp:requiredfieldvalidator></td>
       </tr>
       <tr>
        <td><asp:label id="Label4" runat="server">Re-enter Password:</asp:label></td>
        <td><asp:textbox id="txtPasswordConfirm" runat="server"
TextMode="Password"></asp:textbox></td>
        <td><asp:requiredfieldvalidator id="validatePasswordReEntry" ErrorMessage="You
must re-enter your password." ControlToValidate="txtPasswordConfirm"
          Runat="server"></asp:requiredfieldvalidator>
         <asp:comparevalidator id="validatePasswordMatch" ErrorMessage="You must re-
enter the same password." ControlToValidate="txtPassword"
          Runat="server" Operator="Equal" ControlToCompare="txtPasswordConfirm"
/></td>
       </tr>
       <tr>
        <td><asp:label id="Label5" runat="server">Phone Number:</asp:label></td>
        <td><asp:textbox id="txtPhone" runat="server"></asp:textbox></td>
        <td><asp:requiredfieldvalidator id="validatePhone" ErrorMessage="You must enter a
phone number." ControlToValidate="txtPhone"
          Runat="server"></asp:requiredfieldvalidator></td>
       </tr>
      </table>
     </P>
     <P><asp:button id="btnConfirm" runat="server" Text="Confirm"></asp:button></P>
     <P><asp:label id="lblMsg" runat="server"></asp:label></P>
```

```
    </form>
  </body>
</HTML>
```

Checkout.aspx

```
<%@ Register TagPrefix="uc1" TagName="SearchBox" Src="UserControls/SearchBox.ascx"
%>
<%@ Register TagPrefix="uc1" TagName="CategoriesList"
Src="UserControls/CategoriesList.ascx" %>
<%@ Register TagPrefix="uc1" TagName="Header" Src="UserControls/Header.ascx" %>
<%@ Register TagPrefix="uc1" TagName="DepartmentsList"
Src="UserControls/DepartmentsList.ascx" %>
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="Checkout.aspx.vb"
Inherits="JokePoint.Checkout"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
        <HEAD>
                <title>Checkout</title>
                <meta name="GENERATOR" content="Microsoft Visual Studio .NET
7.1">
                <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
                <meta name="vs_defaultClientScript" content="JavaScript">
                <meta name="vs_targetSchema"
content="http://schemas.microsoft.com/intellisense/ie5">
                <link href="JokePoint.css" type="text/css" rel="stylesheet">
        </HEAD>
        <body>
                <form id="Form1" method="post" runat="server">
<table height="100%" cellSpacing="0" cellPadding="0" width="770" border="0">
        <tr>
        <td vAlign="top" width="190" height="100%">
                <table height="100%" cellSpacing="0" cellPadding="0" width="190"
border="0">
                        <tr>
                        <td vAlign="top" height="100%">
                <uc1:departmentslist id="DepartmentsList1"
runat="server"></uc1:departmentslist>
                <uc1:categorieslist id="CategoriesList1"
runat="server"></uc1:categorieslist>
                <uc1:searchbox id="SearchBox1" runat="server"></uc1:searchbox>
                <br>
<asp:imagebutton id="viewCartButton" runat="server" ImageUrl="Images/ViewCart.gif">
</asp:imagebutton>
                        </td>
                        </tr>
                </table>
                        </td>
                        <td vAlign="top" width="550"><br>
                <table>
```

```
                                <tr>
                                <td>
                                <uc1:Header id="Header1" runat="server"></uc1:Header></td>
                                <tr>
                                <td>
                                <p align="right">
                                <b>User logged in:</b>
                                <asp:label id="txtUserName" runat="server"></asp:label><br>
                                <asp:button id="logOutButton" runat="server" Text="Log
Out"></asp:button><br>
                                <img src="Images/1n.gif" border="0" width="350" height="1">
                                </p>
                                </td>
                                </tr>
                                <tr>
                                <td id="pageContentsCell" runat="server">
                                <P><asp:label id="Label1" runat="server"
CssClass="ListDescription">
                                    Your order consists of the following items:
                                </asp:label></P>
                                <P><asp:datagrid id="grid" runat="server"
AutoGenerateColumns="False" Width="100%">
                <ItemStyle Font-Size="X-Small" Font-Names="Verdana"
BackColor="Gainsboro"></ItemStyle>
                <HeaderStyle Font-Size="X-Small" Font-Names="Verdana" Font-Bold="True"
ForeColor="White" BackColor="Navy"></HeaderStyle>
                <Columns>
                <asp:BoundColumn DataField="Name" ReadOnly="True" HeaderText="Product
Name"></asp:BoundColumn>
                <asp:BoundColumn DataField="Price" ReadOnly="True"
HeaderText="Price"></asp:BoundColumn>
                <asp:BoundColumn DataField="Quantity" ReadOnly="True"
HeaderText="Quantity">
        </asp:BoundColumn>
                <asp:BoundColumn DataField="Subtotal" ReadOnly="True"
HeaderText="Subtotal">
        </asp:BoundColumn>
                </Columns>
                </asp:datagrid></P>
                <P>
                Total amount:
                <asp:label id="totalAmountLabel" runat="server"
CssClass="ProductPrice"></asp:label>
                <br>
                <br>
                <asp:label id="lblCreditCardNote" Runat="server"></asp:label>
                <br>
                <br>
                <asp:label id="lblAddress" Runat="server"></asp:label>
                <br>
                <br>
```

```
                <asp:button id="changeDetailsButton" runat="server" Text="Change Customer
Details"></asp:button>
                <asp:button id="addCreditCardButton" runat="server" Text="Add Credit
Card"></asp:button>
                <asp:button id="addAddressButton" runat="server" Text="Add
Address"></asp:button>
                <br>
                <br>
                <asp:button id="placeOrderButton" runat="server" Text="Place
Order"></asp:button>
                <asp:button id="cancelOrderButton" runat="server" Text="Continue
Shopping"></asp:button>
                </P>
                                </td>
                                </tr>
                                </table>
                                </td>
                                </tr>
                                </table>
                        </form>
                </body>
        </HTML>
```

ordersAdmin.aspx

```
        <%@ Register TagPrefix="uc1" TagName="Logout" Src="AdminUserControls/Logout.ascx"
%>
        <%@ Page Language="vb" AutoEventWireup="false" Codebehind="ordersAdmin.aspx.vb"
Inherits="JokePoint.ordersAdmin"%>
        <%@ Register TagPrefix="uc1" TagName="OrdersAdmin"
Src="AdminUserControls/OrdersAdmin.ascx" %>
        <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
        <HTML>
                <HEAD>
                        <title>ordersAdmin</title>
        <meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.1">
        <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
        <meta name="vs_defaultClientScript" content="JavaScript">
        <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/ie5">
        <link href="JokePoint.css" type="text/css" rel="stylesheet">
        </HEAD>
        <body>
        <form id="Form1" method="post" runat="server">
        <strong><font size="5">Orders Admin Page</font></strong>
        <uc1:Logout id="Logout1" runat="server"></uc1:Logout>
        <br>
        <br>
        <table border="0">
        <tr>
        <td>
```

```
<uc1:OrdersAdmin id="OrdersAdmin1" runat="server"></uc1:OrdersAdmin
</td>
 </tr>
 <tr>
 <td runat="server" id="orderDetailsCell"></td>
 </tr>
 </table>
 </form>
 </body>
 </HTML>
```

OrderDone.aspx

```
<%@ Page %>
<form id="Form1" method="post" runat="server">
  <p>Thank you for your order!</p>
  <p>A confirmation e-mail should arrive shortly.</p>
  <p><a href="default.aspx">Back to shop</a></p>
</form>
```

shoppingCartAdmin.aspx

```
<%@ Page Language="vb" AutoEventWireup="false"
Codebehind="shoppingCartAdmin.aspx.vb" Inherits="JokePoint.shoppingCartAdmin"%>
<%@ Register TagPrefix="uc1" TagName="Logout" Src="AdminUserControls/Logout.ascx"
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
        <HEAD>
                <title>shoppingCartAdmin</title>
                <meta name="GENERATOR" content="Microsoft Visual Studio .NET
7.1">
                <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
                <meta name="vs_defaultClientScript" content="JavaScript">
                <meta name="vs_targetSchema"
content="http://schemas.microsoft.com/intellisense/ie5">
                <link href="JokePoint.css" type="text/css" rel="stylesheet">
        </HEAD>
        <body>
                <form id="Form1" method="post" runat="server">
                        <strong><font size="5">JokePoint Admin Page</font></strong>
                        <uc1:Logout id="Logout1" runat="server"></uc1:Logout>
                        <P>
                                <asp:label id="countLabel" runat="server"
CssClass="AdminPageText">
          Hello!
         </asp:label></P>
                        <P>
```

```
                                        <asp:Label id="Label1" runat="server">How many
days?</asp:Label>
                                        <asp:DropDownList id="daysList" runat="server">
                                                <asp:ListItem Value="1">One</asp:ListItem>
                                                <asp:ListItem Value="10"
Selected="True">Ten</asp:ListItem>
                                                <asp:ListItem
Value="20">Twenty</asp:ListItem>
                                                <asp:ListItem Value="30">Thirty</asp:ListItem>
                                        </asp:DropDownList></P>
                                <P>
        <asp:button id="countButton" runat="server" Text="Count Old Elements"
CssClass="AdminButtonText"></asp:button>
                                </P>
                                <P>
        <asp:button id="deleteButton" runat="server" Text="Delete Old Elements"
CssClass="AdminButtonText"></asp:button>
                                </P>
                        </form>
                </body>
        </HTML>
```

CategoriesAdmin.ascx

```
<%@ Control Language="vb" AutoEventWireup="false" Codebehind="CategoriesAdmin.ascx.vb"
Inherits="JokePoint.CategoriesAdmin" TargetSchema="http://schemas.microsoft.com/intellisense/ie5"
%>
<P>
        <asp:Label id="Label1" runat="server" CssClass="AdminPageText">Edit categories for
department:</asp:Label>
        <asp:LinkButton id="departmentNameLink" runat="server"
CssClass="ListDescription"></asp:LinkButton></P>
<P>
        <asp:DataGrid id="categoriesGrid" runat="server" AutoGenerateColumns="False"
Width="100%">
                <ItemStyle Font-Size="X-Small" Font-Names="verdana"
BackColor="Gainsboro"></ItemStyle>
                <HeaderStyle Font-Size="X-Small" Font-Names="Verdana" Font-Bold="True"
ForeColor="White" BackColor="Navy"></HeaderStyle>
                <Columns>
                        <asp:BoundColumn DataField="Name" HeaderText="Category
Name"></asp:BoundColumn>
                        <asp:TemplateColumn HeaderText="Category Description">
                                <ItemTemplate>
                                        <asp:Label id=Label3 runat="server" Text='<%#
DataBinder.Eval(Container, "DataItem.Description") %>'>
                                        </asp:Label>
                                </ItemTemplate>
                                <EditItemTemplate>
```

```
                                        <asp:TextBox id=editDescriptionTextBox runat="server"
Text='<%# DataBinder.Eval(Container, "DataItem.Description") %>' TextMode="MultiLine"
Width="100%" Rows="3">
                                        </asp:TextBox>
                                    </EditItemTemplate>
                                </asp:TemplateColumn>
                                <asp:EditCommandColumn ButtonType="PushButton"
UpdateText="Update" CancelText="Cancel" EditText="Edit"></asp:EditCommandColumn>
                                <asp:ButtonColumn Text="Edit Products" ButtonType="PushButton"
CommandName="Select"></asp:ButtonColumn>
                                <asp:ButtonColumn Text="Delete" ButtonType="PushButton"
CommandName="Delete"></asp:ButtonColumn>
                        </Columns>
                </asp:DataGrid></P>
<P>
                <asp:Label id="Label2" runat="server" CssClass="AdminPageText">Add new
category:</asp:Label>
                <asp:TextBox id="nameTextBox" runat="server">(category name)</asp:TextBox>
                <asp:TextBox id="descriptionTextBox" runat="server">(category description)</asp:TextBox>
                <asp:Button id="addCategoryButton" runat="server" Text="Add"
CssClass="AdminButtonText"></asp:Button></P>
```

DepartmentsAdmin.ascx

```
<%@ Control Language="vb" AutoEventWireup="false" Codebehind="DepartmentsAdmin.ascx.vb"
Inherits="JokePoint.DepartmentsAdmin"
TargetSchema="http://schemas.microsoft.com/intellisense/ie5" %>
<asp:datagrid id="departmentsGrid" AutoGenerateColumns="False" Width="100%" runat="server">
        <ItemStyle Font-Size="X-Small" Font-Names="Verdana"
BackColor="Gainsboro"></ItemStyle>
        <HeaderStyle Font-Size="X-Small" Font-Names="Verdana" Font-Bold="True"
ForeColor="White" BackColor="Navy"></HeaderStyle>
        <Columns>
                <asp:BoundColumn DataField="Name" HeaderText="Department
Name"></asp:BoundColumn>
                <asp:TemplateColumn HeaderText="Department Description">
                        <ItemTemplate>
                                <asp:Label id=Label3 runat="server" Text='<%#
DataBinder.Eval(Container, "DataItem.Description") %>'>
                                </asp:Label>
                        </ItemTemplate>
                        <EditItemTemplate>
                                <asp:TextBox id=editDescriptionTextBox runat="server"
Text='<%# DataBinder.Eval(Container, "DataItem.Description") %>' TextMode="MultiLine"
Width="100%" Rows="3">
                                </asp:TextBox>
                        </EditItemTemplate>
                </asp:TemplateColumn>
                <asp:EditCommandColumn ButtonType="PushButton" UpdateText="Update"
CancelText="Cancel" EditText="Edit"></asp:EditCommandColumn>
```

```
                <asp:ButtonColumn Text="Edit Categories" ButtonType="PushButton"
CommandName="Select"></asp:ButtonColumn>
                <asp:ButtonColumn Text="Delete" ButtonType="PushButton"
CommandName="Delete"></asp:ButtonColumn>
        </Columns>
</asp:datagrid><br>
        <asp:label id="Label1" runat="server" CssClass="AdminPageText">Add new
department:</asp:label> <asp:textbox id="nameTextBox" runat="server">(department
name)</asp:textbox> <asp:textbox id="descriptionTextBox" runat="server">(department
description)</asp:textbox> <asp:button id="addDepartmentButton" runat="server"
CssClass="AdminButtonText" Text="Add"></asp:button>
```

Logout.ascx

```
<%@ Control Language="vb" AutoEventWireup="false" Codebehind="Logout.ascx.vb"
TargetSchema="http://schemas.microsoft.com/intellisense/ie5" %>
<font class="AdminPageText">(go back to the <a href="default.aspx">storefront</a> or
        <asp:LinkButton id="logoutButton" runat="server">logout
 </asp:LinkButton>) </font>
```

OrderDetailsAdmin.ascx

```
<%@ Control Language="vb" AutoEventWireup="false" Codebehind="OrderDetailsAdmin.ascx.vb"
Inherits="JokePoint.OrderDetailsAdmin"
TargetSchema="http://schemas.microsoft.com/intellisense/ie5" %>
<P>
        <asp:Label id="Label1" runat="server" CssClass="AdminPageText" Width="150px">Order
ID:</asp:Label>
        <asp:TextBox id="orderIdTextBox" runat="server" Width="400px"></asp:TextBox><BR>
        <asp:Label id="Label4" runat="server" CssClass="AdminPageText" Width="150px">Date
Created:</asp:Label>
        <asp:TextBox id="dateCreatedTextBox" runat="server"
Width="400px"></asp:TextBox><BR>
        <asp:Label id="Label5" runat="server" CssClass="AdminPageText" Width="150px">Date
Shipped:</asp:Label>
        <asp:TextBox id="dateShippedTextBox" runat="server"
Width="400px"></asp:TextBox><BR>
        <asp:label id="Label6" CssClass="AdminPageText" Width="150px" runat="server">
    Status:</asp:label>
        <asp:textbox id="statusTextBox" Width="400px" runat="server"
ReadOnly="True"></asp:textbox><BR>
        <asp:label id="Label7" CssClass="AdminPageText" Width="150px" runat="server">
    AuthCode:</asp:label>
        <asp:textbox id="authCodeTextBox" Width="400px" runat="server"
ReadOnly="True"></asp:textbox><BR>
        <asp:label id="Label8" CssClass="AdminPageText" Width="150px" runat="server">
    Reference:</asp:label>
        <asp:textbox id="referenceTextBox" Width="400px" runat="server"
ReadOnly="True"></asp:textbox></P>
```

```
<P>
        <asp:button id="processButton" CssClass="AdminButtonText" Width="302px"
runat="server" Text="Process Button"></asp:button></P>
<P><asp:label id="Label2" CssClass="AdminPageText" Width="150px" runat="server">
    Order Details:</asp:label>
        <asp:DataGrid id="orderDetailsGrid" runat="server" Width="100%"
AutoGenerateColumns="False">
                <ItemStyle Font-Size="X-Small" Font-Names="verdana"
BackColor="Gainsboro"></ItemStyle>
                <HeaderStyle Font-Size="X-Small" Font-Names="verdana" Font-Bold="True"
ForeColor="White" BackColor="Navy"></HeaderStyle>
                <Columns>
                        <asp:BoundColumn DataField="ProductID" HeaderText="Product
ID"></asp:BoundColumn>
                        <asp:BoundColumn DataField="ProductName" HeaderText="Product
Name"></asp:BoundColumn>
                        <asp:BoundColumn DataField="Quantity"
HeaderText="Quantity"></asp:BoundColumn>
                        <asp:BoundColumn DataField="UnitCost" HeaderText="Unit
Cost"></asp:BoundColumn>
                        <asp:BoundColumn DataField="Subtotal"
HeaderText="Subtotal"></asp:BoundColumn>
                </Columns>
        </asp:DataGrid></P>
<P><asp:label id="Label11" CssClass="AdminPageText" Width="150px" runat="server">
    Customer Details:</asp:label>
        <asp:datagrid id="customerGrid" Width="100%" runat="server"
AutoGenerateColumns="False">
                <ItemStyle Font-Size="X-Small" Font-Names="Verdana"
BackColor="Gainsboro"></ItemStyle>
                <HeaderStyle Font-Size="X-Small" Font-Names="Verdana" Font-Bold="True"
ForeColor="White" BackColor="Navy"></HeaderStyle>
                <Columns>
                        <asp:BoundColumn DataField="CustomerID"
HeaderText="ID"></asp:BoundColumn>
                        <asp:BoundColumn DataField="Name"
HeaderText="Name"></asp:BoundColumn>
                        <asp:BoundColumn DataField="Email"
HeaderText="EMail"></asp:BoundColumn>
                        <asp:BoundColumn DataField="Address1"
HeaderText="Address1"></asp:BoundColumn>
                        <asp:BoundColumn DataField="Address2"
HeaderText="Address2"></asp:BoundColumn>
                        <asp:BoundColumn DataField="City"
HeaderText="Town/City"></asp:BoundColumn>
                        <asp:BoundColumn DataField="Region"
HeaderText="Region/State"></asp:BoundColumn>
                        <asp:BoundColumn DataField="PostalCode" HeaderText="Postal
Code/ZIP"></asp:BoundColumn>
                        <asp:BoundColumn DataField="Country"
HeaderText="Country"></asp:BoundColumn>
```

```
                    <asp:BoundColumn DataField="Phone"
HeaderText="Phone"></asp:BoundColumn>
                </Columns>
        </asp:datagrid></P>
<P><asp:label id="Label3" CssClass="AdminPageText" Width="150px" runat="server">
    Audit Trail:</asp:label>
        <asp:datagrid id="auditGrid" Width="100%" runat="server" AutoGenerateColumns="False">
                <ItemStyle Font-Size="X-Small" Font-Names="Verdana"
BackColor="Gainsboro"></ItemStyle>
                <HeaderStyle Font-Size="X-Small" Font-Names="Verdana" Font-Bold="True"
ForeColor="White" BackColor="Navy"></HeaderStyle>
                <Columns>
                    <asp:BoundColumn DataField="DateStamp" HeaderText="Date
Recorded"></asp:BoundColumn>
                    <asp:BoundColumn DataField="Message"
HeaderText="Message"></asp:BoundColumn>
                    <asp:BoundColumn DataField="MessageNumber" HeaderText="Message
Number"></asp:BoundColumn>
                </Columns>
        </asp:datagrid></P>
```

OrdersAdmin.ascx

```
<%@ Control Language="vb" AutoEventWireup="false" Codebehind="OrdersAdmin.ascx.vb"
TargetSchema="http://schemas.microsoft.com/intellisense/ie5" %>
<P>
 <asp:Label id="Label1" runat="server" CssClass="AdminPageText">Show the most recent
</asp:Label>
 <asp:TextBox id="recordCountTextBox" runat="server" Width="35px">20</asp:TextBox>
 <asp:Label id="Label2" runat="server" CssClass="AdminPageText">orders</asp:Label>
 <asp:Button id="ordersByRecentButton" runat="server" Text="Go!"
CssClass="AdminButtonText"></asp:Button><BR>
 <asp:Label id="Label3" runat="server" CssClass="AdminPageText">Show all orders created between
</asp:Label>
 <asp:TextBox id="startDateTextBox" runat="server" Width="70px"></asp:TextBox>
 <asp:Label id="Label4" runat="server" CssClass="AdminPageText">and</asp:Label>
 <asp:TextBox id="endDateTextBox" runat="server" Width="70px"></asp:TextBox>
 <asp:button id="ordersByDateButton" CssClass="AdminButtonText" runat="server"
Text="Go!"></asp:button><BR>
 <asp:label id="Label5" CssClass="AdminPageText" runat="server">
    Show orders by status</asp:label>
 <asp:dropdownlist id="statusList" Runat="server"></asp:dropdownlist>
 <asp:button id="ordersByStatusButton" CssClass="AdminButtonText" runat="server"
Text="Go!"></asp:button><BR>
 <asp:label id="Label6" CssClass="AdminPageText" runat="server">
    Show orders for customer with CustomerID</asp:label>
 <asp:textbox id="customerIDTextBox" runat="server" Width="35px">
    1</asp:textbox>
 <asp:button id="ordersByCustomerButton" CssClass="AdminButtonText" runat="server"
Text="Go!"></asp:button><BR>
```

```
 <asp:label id="Label7" CssClass="AdminPageText" runat="server">
    Show order with OrderID</asp:label>
 <asp:textbox id="orderIDTextBox" runat="server" Width="35px">
    1</asp:textbox>
 <asp:button id="orderByIDButton" CssClass="AdminButtonText" runat="server"
Text="Go!"></asp:button></P>
<P>
 <asp:Label id="errorLabel" runat="server" CssClass="AdminErrorText"
EnableViewState="False"></asp:Label>
 <asp:RangeValidator id="startDateValidator" runat="server" ErrorMessage="Invalid start date."
ControlToValidate="startDateTextBox"
   Display="None" MaximumValue="1/1/2005" MinimumValue="1/1/1999"
Type="Date"></asp:RangeValidator>
 <asp:RangeValidator id="endDateValidator" runat="server" ErrorMessage="Invalid end date."
ControlToValidate="endDateTextBox"
   Display="None" MaximumValue="1/1/2005" MinimumValue="1/1/1999"
Type="Date"></asp:RangeValidator>
 <asp:requiredfieldvalidator id="customerIDValidator" Display="None"
ControlToValidate="customerIDTextBox" ErrorMessage="You must enter a customer ID."
   Runat="server"></asp:requiredfieldvalidator>
 <asp:requiredfieldvalidator id="orderIDValidator" Display="None"
ControlToValidate="orderIDTextBox" ErrorMessage="You must enter an order ID."
   Runat="server"></asp:requiredfieldvalidator>
 <asp:CompareValidator id="compareDatesValidator" runat="server" ErrorMessage="End date should
be more recent than start date."
   ControlToValidate="startDateTextBox" Display="None" Type="Date"
ControlToCompare="endDateTextBox" Operator="LessThan"></asp:CompareValidator>
 <asp:ValidationSummary id="validationSummary" runat="server" CssClass="AdminErrorText"
HeaderText="Validation errors:"></asp:ValidationSummary><BR>
 <asp:DataGrid id="grid" runat="server" Width="100%" AutoGenerateColumns="False">
   <ItemStyle Font-Size="X-Small" Font-Names="Verdana" BackColor="Gainsboro"></ItemStyle>
          <HeaderStyle Font-Size="X-Small" Font-Names="Verdana" Font-Bold="True"
ForeColor="White"
BackColor="Navy"></HeaderStyle>
   <Columns>
    <asp:BoundColumn DataField="OrderID" HeaderText="Order ID"></asp:BoundColumn>
    <asp:BoundColumn DataField="DateCreated" HeaderText="Date Created"></asp:BoundColumn>
    <asp:BoundColumn DataField="DateShipped" HeaderText="Date Shipped"></asp:BoundColumn>
    <asp:BoundColumn DataField="Description" HeaderText="Status"></asp:BoundColumn>
    <asp:BoundColumn DataField="CustomerID" HeaderText="Customer ID"></asp:BoundColumn>
    <asp:BoundColumn DataField="Name" HeaderText="Customer Name"></asp:BoundColumn>
          <asp:ButtonColumn Text="View Details" ButtonType="PushButton"
CommandName="Select"></asp:ButtonColumn>
   </Columns>
 </asp:DataGrid></P>
<P></P>
```

ProductDetailsAdmin.ascx

```
<%@ Control Language="vb" AutoEventWireup="false" Codebehind="ProductDetailsAdmin.ascx.vb"
TargetSchema="http://schemas.microsoft.com/intellisense/ie5" %>
<table>
        <tr>
                <td width="150">
                        <P align="center">
                                <asp:Image id="productImage" runat="server"></asp:Image></P>
                </td>
                <td>
                        <P>
                                <asp:Label id="Label1" runat="server"
CssClass="AdminPageText">Editing product:</asp:Label>
                                <asp:LinkButton id="productNameLink" runat="server"
CssClass="ListDescription"></asp:LinkButton></P>
                        <P>
                                <asp:Label id="Label2" runat="server"
CssClass="AdminPageText">Product belongs to these categories:</asp:Label>
                                <asp:Label id="categoriesListLabel"
runat="server"></asp:Label></P>
                        <P>
<asp:Label id="Label4" runat="server" CssClass="AdminPageText">Assign product to this
category:</asp:Label>
                                <asp:DropDownList id="categoriesList"
runat="server"></asp:DropDownList>
                                <asp:Button id="assignButton" runat="server" Text="Assign"
CssClass="AdminButtonText"></asp:Button></P>
                        <P>
                                <asp:Label id="Label5" runat="server"
CssClass="AdminPageText">Move product to this category:</asp:Label>
                                <asp:DropDownList id="categoriesList2"
runat="server"></asp:DropDownList>
                                <asp:Button id="moveButton" runat="server" Text="Move"
CssClass="AdminButtonText"></asp:Button></P>
                        <P>
                                <asp:Button id="removeButton" runat="server" Text="Button"
CssClass="AdminButtonText"></asp:Button></P>
                        Upload picture: <input id="fileName" type="file" name="fileName"
runat="server">
                        <asp:button id="uploadFile" text="Upload" runat="server"></asp:button>
                        <asp:label id="uploadMessageLabel" runat="server"></asp:label>
                </td>
        </tr>
</table>
```

ProductsAdmin.ascx

```
<%@ Control Language="vb" AutoEventWireup="false"
Codebehind="ProductsAdmin.ascx.vb"TargetSchema="http://schemas.microsoft.com/intellisense/ie5"
%>
<P>
```

```
        <asp:Label id="firstLabel" CssClass="AdminPageText" runat="server">Editing products for
category:</asp:Label>
        <asp:LinkButton id="categoryNameLink" CssClass="ListDescription"
runat="server"></asp:LinkButton></P>
<P>
        <asp:DataGrid id="productsGrid" runat="server" AutoGenerateColumns="False">
                <ItemStyle Font-Size="X-Small" Font-Names="verdana"
BackColor="Gainsboro"></ItemStyle>
                <HeaderStyle Font-Size="X-Small" Font-Names="verdana" Font-Bold="True"
ForeColor="White" BackColor="Navy"></HeaderStyle>
                <Columns>
                        <asp:TemplateColumn HeaderText="Product Image">
                                <ItemTemplate>
                                        <img border="0" height="50" src='ProductImages/<%#
DataBinder.Eval(Container, "DataItem.ImagePath") %>'>
                                </ItemTemplate>
                        </asp:TemplateColumn>
                        <asp:BoundColumn DataField="Name" HeaderText="Product
Name"></asp:BoundColumn>
                        <asp:TemplateColumn HeaderText="Product Description">
                                <ItemTemplate>
                                        <asp:Label runat="server" Text='<%#
DataBinder.Eval(Container, "DataItem.Description") %>'>
                                        </asp:Label>
                                </ItemTemplate>
                                <EditItemTemplate>
                                        <asp:TextBox id=editDescriptionTextBox runat="server"
Text='<%# DataBinder.Eval(Container, "DataItem.Description") %>' TextMode="MultiLine"
Width="250" Rows="3">
                                        </asp:TextBox>
                                </EditItemTemplate>
                        </asp:TemplateColumn>
                        <asp:BoundColumn DataField="Price"
HeaderText="Price"></asp:BoundColumn>
                        <asp:BoundColumn DataField="ImagePath" HeaderText="Image
Path"></asp:BoundColumn>
                        <asp:TemplateColumn HeaderText="Dept.prom.">
                                <ItemTemplate>
                                        <asp:CheckBox Text="" Enabled="False" runat="server"
ID="Checkbox1" Checked='<%# DataBinder.Eval(Container, "DataItem.OnDepartmentPromotion")
%>' >
                                        </asp:CheckBox>
                                </ItemTemplate>
                                <EditItemTemplate>
                                        <asp:CheckBox id="deptPromListCheck" Text=""
runat="server" Checked='<%# DataBinder.Eval(Container, "DataItem.OnDepartmentPromotion") %>'>
                                        </asp:CheckBox>
                                </EditItemTemplate>
                        </asp:TemplateColumn>
                        <asp:TemplateColumn HeaderText="Cat.prom.">
                                <ItemTemplate>
```

```
                                        <asp:CheckBox Text="" Enabled="False" runat="server"
Checked='<%# DataBinder.Eval(Container, "DataItem.OnCatalogPromotion") %>' ID="Checkbox2">
                                        </asp:CheckBox>
                                    </ItemTemplate>
                                    <EditItemTemplate>
                                        <asp:CheckBox id="catPromListCheck" Text=""
runat="server" Checked='<%# DataBinder.Eval(Container, "DataItem.OnCatalogPromotion") %>'>
                                        </asp:CheckBox>
                                    </EditItemTemplate>
                                </asp:TemplateColumn>
                                <asp:EditCommandColumn ButtonType="PushButton"
UpdateText="Update" CancelText="Cancel" EditText="Edit"></asp:EditCommandColumn>
                                <asp:ButtonColumn Text="Select" ButtonType="PushButton"
CommandName="Select"></asp:ButtonColumn>
                        </Columns>
                </asp:DataGrid></P>
<P>
                <asp:Label id="newProductLabel" CssClass="AdminPageText" runat="server">Create a new
product and add it to the selected category:</asp:Label><BR>
                <asp:TextBox id="nameTextBox" runat="server">(name)</asp:TextBox>
                <asp:TextBox id="descriptionTextBox" runat="server">(description)</asp:TextBox>
                <asp:TextBox id="priceTextBox" runat="server">(price)</asp:TextBox>
                <asp:CheckBox id="departmentPromotionCheck" CssClass="AdminPageText" runat="server"
Text="Department Promotion"></asp:CheckBox>
                <asp:CheckBox id="catalogPromotionCheck" CssClass="AdminPageText" runat="server"
Text="Catalog Promotion"></asp:CheckBox>
                        <asp:Button id="createProductButton" CssClass="AdminButtonText" runat="server"
Text="Add"></asp:Button> </P>
```

APPENDIX B

VISUAL BASIC .NET FILE

Catalog.vb

```
Imports System.Data.SqlClient

Public Class DepartmentDetails
    Public Name As String
    Public Description As String
End Class

Public Class CategoryDetails
    Public Name As String
    Public Description As String
End Class

Public Class Catalog
    Public Shared Function GetDepartments() As SqlDataReader
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetDepartments", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Open the connection
        connection.Open()
        ' Return a SqlDataReader to the calling function
        Return command.ExecuteReader(CommandBehavior.CloseConnection)
    End Function

    Public Shared Function GetDepartmentDetails(ByVal departmentId As String) As
DepartmentDetails
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetDepartmentDetails", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@DepartmentID", SqlDbType.Int, 4)
        command.Parameters("@DepartmentID").Value = departmentId
        ' Add an output parameter
        command.Parameters.Add("@DepartmentName", SqlDbType.VarChar, 50)
        command.Parameters("@DepartmentName").Direction = ParameterDirection.Output
        ' Add an output parameter
        command.Parameters.Add("@DepartmentDescription", SqlDbType.VarChar, 200)
        command.Parameters("@DepartmentDescription").Direction = ParameterDirection.Output
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
        ' Populate a DepartmentDetails object with data from output parameters
        Dim details As New DepartmentDetails()
```

```
        details.Name = command.Parameters("@DepartmentName").Value.ToString()
        details.Description = command.Parameters("@DepartmentDescription").Value.ToString()
        ' Return the DepartmentDetails object to the calling function
        Return details
    End Function

    Public Shared Function GetCategoriesInDepartment(ByVal departmentId As String) As
SqlDataReader
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetCategoriesInDepartment", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@DepartmentID", SqlDbType.Int, 4)
        command.Parameters("@DepartmentID").Value = departmentId
        Try
            ' Open the connection
            connection.Open()
            ' Return an SqlDataReader to the calling function
            Return command.ExecuteReader(CommandBehavior.CloseConnection)
        Catch e As Exception
            ' Close the connection and throw the exception
            connection.Close()
            Throw e
        End Try
    End Function

    Public Shared Function GetCategoryDetails(ByVal categoryId As String) As
CategoryDetails
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetCategoryDetails", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CategoryID", SqlDbType.Int, 4)
        command.Parameters("@CategoryID").Value = categoryId
        ' Add an output parameter
        command.Parameters.Add("@CategoryName", SqlDbType.VarChar, 50)
        command.Parameters("@CategoryName").Direction = ParameterDirection.Output
        ' Add an output parameter
        command.Parameters.Add("@CategoryDescription", SqlDbType.VarChar, 200)
        command.Parameters("@CategoryDescription").Direction = ParameterDirection.Output
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
        ' Populate a CategoryDetails object with data from output parameters
```

```
        Dim details As New CategoryDetails()
        details.Name = command.Parameters("@CategoryName").Value.ToString()
        details.Description = command.Parameters("@CategoryDescription").Value.ToString()
        ' Return the CategoryDetails object to the calling function
        Return details
    End Function

    Public Shared Function GetProductsInCategory(ByVal categoryId As String) As
SqlDataReader
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetProductsInCategory", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CategoryID", SqlDbType.Int, 4)
        command.Parameters("@CategoryID").Value = categoryId
        Try
            ' Open the connection
            connection.Open()
            ' Return an SqlDataReader to the calling function
            Return command.ExecuteReader(CommandBehavior.CloseConnection)
        Catch e As Exception
            ' Close the connection and throw the exception
            connection.Close()
            Throw e
        End Try
    End Function

    Public Shared Function GetProductsOnDepartmentPromotion(ByVal departmentId As String)
As SqlDataReader
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetProductsOnDepartmentPromotion", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@DepartmentID", SqlDbType.Int, 4)
        command.Parameters("@DepartmentID").Value = departmentId
        Try
            ' Open the connection
            connection.Open()
            ' Return an SqlDataReader to the calling function
            Return command.ExecuteReader(CommandBehavior.CloseConnection)
        Catch e As Exception
            ' Close the connection and throw the exception
            connection.Close()
            Throw e
        End Try
    End Function

        Public Shared Function GetProductsOnCatalogPromotion() As SqlDataReader
```

```
    ' Create the connection object
    Dim connection As New SqlConnection(connectionString)
    ' Create and initialize the command object
    Dim command As New SqlCommand("GetProductsOnCatalogPromotion", connection)
    command.CommandType = CommandType.StoredProcedure

    Try
        ' Open the connection
        connection.Open()
        ' Return an SqlDataReader to the calling function
        Return command.ExecuteReader(CommandBehavior.CloseConnection)
    Catch e As Exception
        ' Close the connection and throw the exception
        connection.Close()
        Throw e
    End Try
End Function

Public Shared Function SearchCatalog(ByVal searchString As String, _
                ByVal pageNumber As String, _
                ByVal productsOnPage As String, _
                ByVal allWords As String) _
                As Integer
    ' Create the connection object
    Dim connection As New SqlConnection(connectionString)

    ' Create and initialize the command object
    Dim command As New SqlCommand("SearchCatalog", connection)
    command.CommandType = CommandType.StoredProcedure

    ' Add the @AllWords parameter
    ' Guard against bogus values here - if you receive anything
    ' different than "TRUE" assume it's "FALSE"
    If allWords.ToUpper = "TRUE" Then
        ' only do an "all words" search
        command.Parameters.Add("@AllWords", SqlDbType.Bit)
        command.Parameters("@AllWords").Value = 1
    Else
        ' only do an "any words" search
        command.Parameters.Add("@AllWords", SqlDbType.Bit)
        command.Parameters("@AllWords").Value = 0
    End If

    ' Add the @PageNumber parameter
    command.Parameters.Add("@PageNumber", SqlDbType.TinyInt)
    command.Parameters("@PageNumber").Value = pageNumber

    ' Add the @ProductsOnPage parameter
    command.Parameters.Add("@ProductsOnPage", SqlDbType.TinyInt)
    command.Parameters("@ProductsOnPage").Value = productsOnPage

    ' Add the @HowManyResults output parameter
```

```
        command.Parameters.Add("@HowManyResults", SqlDbType.SmallInt)
        command.Parameters("@HowManyResults").Direction = ParameterDirection.Output

        ' Eliminate separation characters
        searchString = searchString.Replace(",", " ")
        searchString = searchString.Replace(";", " ")
        searchString = searchString.Replace(".", " ")
        searchString = searchString.Replace("!", " ")
        searchString = searchString.Replace("?", " ")
        searchString = searchString.Replace("-", " ")

        ' Create an array that contains the words
        Dim words() As String = Split(searchString, " ")
        ' wordsCount contains the total number of words in the array
        Dim wordsCount As Integer = words.Length
        ' index is used to parse the list of words
        Dim index As Integer = 0
        ' this will store the total number of added words
        Dim addedWords As Integer = 0

        ' Allow a maximum of five words
        While addedWords < 5 And index < wordsCount
            ' Add the @WordN parameters here
            ' Only add words having more than two letters
            If Len(words(index)) > 2 Then
                addedWords += 1
                ' Add an input parameter and supply a value for it
                command.Parameters.Add("@Word" + addedWords.ToString, words(index))
            End If
            index += 1
        End While

        ' Time to execute the command
        Try
            ' Open the connection
            connection.Open()
            ' Create and initialize an SqlDataReader object
            Dim reader As SqlDataReader
            reader = command.ExecuteReader(CommandBehavior.CloseConnection)
            ' Store the search results to a DataTable
            Dim table As New DataTable()
            ' Copy column information from the SqlDataReader to the DataTable
            Dim fieldCount As Integer = reader.FieldCount
            Dim fieldIndex As Integer
            For fieldIndex = 0 To fieldCount - 1
                table.Columns.Add(reader.GetName(fieldIndex),
reader.GetFieldType(fieldIndex))
            Next
            ' Copy data from the SqlDataReader to the DataTable
            Dim row As DataRow
            While reader.Read()
                row = table.NewRow()
```

```
            For fieldIndex = 0 To fieldCount - 1
                row(fieldIndex) = reader(fieldIndex)
            Next
            table.Rows.Add(row)
        End While
        ' Close the reader and return the number of results
        reader.Close()

        ' Save the search results to the current session
        HttpContext.Current.Session("SearchTable") = table

        ' return the total number of matching products
        Return command.Parameters("@HowManyResults").Value
    Catch e As Exception
        ' Close the connection and throw the exception
        connection.Close()
        Throw e
    End Try
End Function


Private Shared ReadOnly Property connectionString() As String
    Get
        Return ConfigurationSettings.AppSettings("ConnectionString")
    End Get
End Property
End Class
```

CatalogAdmin

```
Imports System.Data.SqlClient
Public Class CatalogAdmin
#Region "  Departments  "
    Public Shared Function GetDepartmentsWithDescriptions() As SqlDataReader
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetDepartmentsWithDescriptions", connection)
        command.CommandType = CommandType.StoredProcedure
        Try
            ' Open the connection
            connection.Open()
            ' Return an SqlDataReader to the calling function
            Return command.ExecuteReader(CommandBehavior.CloseConnection)
        Catch e As Exception
            ' Close the connection and throw the exception
            connection.Close()
            Throw e
        End Try
```

```
    End Function

    Public Shared Function UpdateDepartment(ByVal departmentId As String, ByVal
departmentName As String, ByVal departmentDescription As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("UpdateDepartment", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@DepartmentID", SqlDbType.Int)
        command.Parameters("@DepartmentID").Value = departmentId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@DepartmentName", SqlDbType.VarChar, 50)
        command.Parameters("@DepartmentName").Value = departmentName
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@DepartmentDescription", SqlDbType.VarChar, 200)
        command.Parameters("@DepartmentDescription").Value = departmentDescription
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function

    Public Shared Function DeleteDepartment(ByVal departmentId As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("DeleteDepartment", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@DepartmentID", SqlDbType.Int)
        command.Parameters("@DepartmentID").Value = departmentId
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function

    Public Shared Function AddDepartment(ByVal departmentName As String, ByVal
departmentDescription As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("AddDepartment", connection)
        command.CommandType = CommandType.StoredProcedure
```

```
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@DepartmentName", SqlDbType.VarChar, 50)
        command.Parameters("@DepartmentName").Value = departmentName
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@DepartmentDescription", SqlDbType.VarChar, 200)
        command.Parameters("@DepartmentDescription").Value = departmentDescription
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function
#End Region

#Region "  Categories  "
    Public Shared Function GetCategoriesWithDescriptions( _
                          ByVal departmentId As String) As SqlDataReader
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetCategoriesWithDescriptions", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@DepartmentID", SqlDbType.Int)
        command.Parameters("@DepartmentID").Value = departmentId
        Try
            ' Open the connection
            connection.Open()
            ' Return an SqlDataReader to the calling function
            Return command.ExecuteReader(CommandBehavior.CloseConnection)
        Catch e As Exception
            ' Close the connection and throw the exception
            connection.Close()
            Throw e
        End Try
    End Function

    Public Shared Function UpdateCategory(ByVal categoryId As String, _
              ByVal categoryName As String, ByVal categoryDescription As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("UpdateCategory", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CategoryID", SqlDbType.Int)
        command.Parameters("@CategoryID").Value = categoryId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CategoryName", SqlDbType.VarChar, 50)
        command.Parameters("@CategoryName").Value = categoryName
```

```
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CategoryDescription", SqlDbType.VarChar, 200)
        command.Parameters("@CategoryDescription").Value = categoryDescription
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function

    Public Shared Function DeleteCategory(ByVal categoryId As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("DeleteCategory", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CategoryID", SqlDbType.Int)
        command.Parameters("@CategoryID").Value = categoryId
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function

    Public Shared Function AddCategory(ByVal departmentID As String, _
      ByVal categoryName As String, ByVal categoryDescription As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("AddCategory", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@DepartmentID", SqlDbType.Int)
        command.Parameters("@DepartmentID").Value = departmentID
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CategoryName", SqlDbType.VarChar, 50)
        command.Parameters("@CategoryName").Value = categoryName
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CategoryDescription", SqlDbType.VarChar, 200)
        command.Parameters("@CategoryDescription").Value = categoryDescription
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
```

```
        End Try
            End Function

#End Region

#Region "  Products  "
    Public Shared Function CreateProductToCategory(ByVal categoryId As String, _
        ByVal productName As String, ByVal productDescription As String, _
        ByVal productPrice As String, ByVal productImage As String, _
        ByVal onDepartmentPromotion As String, _
        ByVal onCatalogPromotion As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("CreateProductToCategory", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CategoryID", SqlDbType.Int)
        command.Parameters("@CategoryID").Value = categoryId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductName", SqlDbType.VarChar, 50)
        command.Parameters("@ProductName").Value = productName
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductDescription", SqlDbType.VarChar, 1000)
        command.Parameters("@ProductDescription").Value = productDescription
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductPrice", SqlDbType.Money, 8)
        command.Parameters("@ProductPrice").Value = productPrice
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductImage", SqlDbType.VarChar, 50)
        command.Parameters("@ProductImage").Value = productImage
        ' Add an input parameter and supply a value for it
        If onDepartmentPromotion.ToUpper = "TRUE" Or onDepartmentPromotion = "1" Then
            ' If we receive "True" or "1" for onDepartmentPromotion ...
            command.Parameters.Add("@OnDepartmentPromotion", SqlDbType.Bit, 1)
            command.Parameters("@OnDepartmentPromotion").Value = 1
        Else
            ' For all the other values we assume the product is not on promotion
            command.Parameters.Add("@OnDepartmentPromotion", SqlDbType.Bit, 1)
            command.Parameters("@OnDepartmentPromotion").Value = 0
        End If
        ' Add an input parameter and supply a value for it
        If onCatalogPromotion.ToUpper = "TRUE" Or onCatalogPromotion = "1" Then
            ' If we receive "True" or "1" for onCatalogPromotion ...
            command.Parameters.Add("@OnCatalogPromotion", SqlDbType.Bit, 1)
            command.Parameters("@OnCatalogPromotion").Value = 1
        Else
            ' For all the other values we assume the product is not on promotion
            command.Parameters.Add("@OnCatalogPromotion", SqlDbType.Bit, 1)
            command.Parameters("@OnCatalogPromotion").Value = 0
        End If
        ' Open the connection, execute the command, and close the connection
```

```
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function

    ' This is identical to CreateProductToCategory, except that we call
    ' the UpdateProduct stored procedure instead of CreateProductToCategory
    Public Shared Function UpdateProduct(ByVal productId As String, _
        ByVal productName As String, ByVal productDescription As String, _
        ByVal productPrice As String, ByVal productImage As String, _
        ByVal onDepartmentPromotion As String, _
        ByVal onCatalogPromotion As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("UpdateProduct", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductID", SqlDbType.Int)
        command.Parameters("@ProductID").Value = productId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductName", SqlDbType.VarChar, 50)
        command.Parameters("@ProductName").Value = productName
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductDescription", SqlDbType.VarChar, 1000)
        command.Parameters("@ProductDescription").Value = productDescription
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductPrice", SqlDbType.Money, 8)
        command.Parameters("@ProductPrice").Value = productPrice
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductImage", SqlDbType.VarChar, 50)
        command.Parameters("@ProductImage").Value = productImage
        If onDepartmentPromotion.ToUpper = "TRUE" Or onDepartmentPromotion = "1" Then
            ' If we receive "True" or "1" for onDepartmentPromotion ...
            command.Parameters.Add("@OnDepartmentPromotion", SqlDbType.Bit, 1)
            command.Parameters("@OnDepartmentPromotion").Value = 1
        Else
            ' For all the other values we assume the product is not on promotion
            command.Parameters.Add("@OnDepartmentPromotion", SqlDbType.Bit, 1)
            command.Parameters("@OnDepartmentPromotion").Value = 0
        End If
        If onCatalogPromotion.ToUpper = "TRUE" Or onCatalogPromotion = "1" Then
            ' If we receive "True" or "1" for onCatalogPromotion ...
            command.Parameters.Add("@OnCatalogPromotion", SqlDbType.Bit, 1)
            command.Parameters("@OnCatalogPromotion").Value = 1
        Else
            ' For all the other values we assume the product is not on promotion
            command.Parameters.Add("@OnCatalogPromotion", SqlDbType.Bit, 1)
            command.Parameters("@OnCatalogPromotion").Value = 0
```

```
        End If
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function
#End Region

#Region " Product Details "
    Public Shared Function RemoveFromCategoryOrDeleteProduct(ByVal productId As String,
ByVal categoryId As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("RemoveFromCategoryOrDeleteProduct", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductID", SqlDbType.Int)
        command.Parameters("@ProductID").Value = productId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CategoryID", SqlDbType.Int)
        command.Parameters("@CategoryID").Value = categoryId
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function

    Public Shared Function GetCategoriesForProduct(ByVal productId As String) As
SqlDataReader
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetCategoriesForProduct", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductID", SqlDbType.Int)
        command.Parameters("@ProductID").Value = productId
        Try
            ' Open the connection
            connection.Open()
            ' Return an SqlDataReader to the calling function
            Return command.ExecuteReader(CommandBehavior.CloseConnection)
        Catch e As Exception
            ' Close the connection and throw the exception
            connection.Close()
```

```
            Throw e
        End Try
    End Function

    Public Shared Function GetCategoriesForNotProduct(ByVal productId As String) As
SqlDataReader
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetCategoriesForNotProduct", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductID", SqlDbType.Int)
        command.Parameters("@ProductID").Value = productId
        Try
            ' Open the connection
            connection.Open()
            ' Return an SqlDataReader to the calling function
            Return command.ExecuteReader(CommandBehavior.CloseConnection)
        Catch e As Exception
            ' Close the connection and throw the exception
            connection.Close()
            Throw e
        End Try
    End Function

    Public Shared Function AssignProductToCategory(ByVal productId As String, ByVal
categoryId As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("AssignProductToCategory", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductID", SqlDbType.Int, 4)
        command.Parameters("@ProductID").Value = productId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CategoryID", SqlDbType.Int, 4)
        command.Parameters("@CategoryID").Value = categoryId
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function

    Public Shared Function MoveProductToCategory(ByVal productId As String, ByVal
oldCategoryId As String, ByVal newCategoryID As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
```

```
        ' Create and initialize the command object
        Dim command As New SqlCommand("MoveProductToCategory", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductID", SqlDbType.Int, 4)
        command.Parameters("@ProductID").Value = productId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@OldCategoryID", SqlDbType.Int, 4)
        command.Parameters("@OldCategoryID").Value = oldCategoryId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@NewCategoryID", SqlDbType.Int, 4)
        command.Parameters("@NewCategoryID").Value = newCategoryID
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function

    Public Shared Function UpdateProductPicture(ByVal productId As String, ByVal
imageFileName As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("UpdateProductPicture", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductID", SqlDbType.Int)
        command.Parameters("@ProductID").Value = productId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ImageFileName", SqlDbType.VarChar, 50)
        command.Parameters("@ImageFileName").Value = imageFileName
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function
#End Region

    Private Shared ReadOnly Property connectionString() As String
        Get
            Return ConfigurationSettings.AppSettings("ConnectionString")
        End Get
    End Property
End Class
```

ShoppingCart.vb

```
Imports System.Data.SqlClient

Public Class ShoppingCart
    Private Shared ReadOnly Property shoppingCartId()
        Get
            Dim context As HttpContext = HttpContext.Current

            ' If the JokePoint_CartID cookie doesn't exit
            ' on client machine we create it with a new GUID
            If context.Request.Cookies("JokePoint_CartID") Is Nothing Then
                ' Generate a new GUID
                Dim cartId As Guid = Guid.NewGuid()
                ' Create the cookie and set its value
                Dim cookie As New HttpCookie("JokePoint_CartID", cartId.ToString)
                ' Current Date
                Dim currentDate As DateTime = DateTime.Now()
                ' Set the time span to 10 days
                Dim ts As New TimeSpan(10, 0, 0, 0)
                ' Expiration Date
                Dim expirationDate As DateTime = currentDate.Add(ts)
                ' Set the Expiration Date to the cookie
                cookie.Expires = expirationDate
                ' Set the cookie on client's browser
                context.Response.Cookies.Add(cookie)
                ' return the Cart ID
                Return cartId.ToString()
            Else
                ' return the value stored in JokePoint_CartID
                Return context.Request.Cookies("JokePoint_CartID").Value
            End If
        End Get
    End Property

    Private Shared ReadOnly Property connectionString() As String
        Get
            Return ConfigurationSettings.AppSettings("ConnectionString")
        End Get
    End Property

    Public Shared Function AddProduct(ByVal productId As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("AddProductToCart", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CartID", SqlDbType.Char, 36)
        command.Parameters("@CartID").Value = shoppingCartId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductID", SqlDbType.Int)
```

```
        command.Parameters("@ProductID").Value = productId
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function

    Public Shared Function UpdateProductQuantity(ByVal productId As String, ByVal quantity
As Integer)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("UpdateCartItem", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CartID", SqlDbType.Char, 36)
        command.Parameters("@CartID").Value = shoppingCartId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductID", SqlDbType.Int)
        command.Parameters("@ProductID").Value = productId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@Quantity", SqlDbType.Int)
        command.Parameters("@Quantity").Value = quantity
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function

    Public Shared Function RemoveProduct(ByVal productId As String)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("RemoveProductFromCart", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CartID", SqlDbType.Char, 36)
        command.Parameters("@CartID").Value = shoppingCartId
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@ProductID", SqlDbType.Int)
        command.Parameters("@ProductID").Value = productId
        ' Open the connection, execute the command, and close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
```

```
            connection.Close()
        End Try
    End Function

    Public Shared Function GetProducts() As SqlDataReader
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetShoppingCartProducts", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CartID", SqlDbType.Char, 36)
        command.Parameters("@CartID").Value = shoppingCartId
        Try
            ' Open the connection and return the results
            connection.Open()
            Return command.ExecuteReader(CommandBehavior.CloseConnection)
        Catch e As Exception
            ' Close the connection and throw the exception
            connection.Close()
            Throw e
        End Try
    End Function

    Public Shared Function GetTotalAmount() As Decimal
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As SqlCommand = New SqlCommand("GetTotalAmount", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CartID", SqlDbType.Char, 36)
        command.Parameters("@CartID").Value = shoppingCartId
        ' The amount has a default value of 0
        Dim amount As Decimal = 0
        Try
            ' Try to get the amount from the database
            connection.Open()
            amount = command.ExecuteScalar()
        Finally
            ' Close the connection
            connection.Close()
        End Try
        ' Return the amount
        Return amount
    End Function

    ' Removes ViewCart parameters from the query string
    Public Shared Function RemoveCartFromQueryString() As String
        ' Will contain the query string without the ViewCart parameters
        Dim newQueryString As String = String.Empty
        Try
```

```
            ' query stores the current query string
            Dim query As System.Collections.Specialized.NameValueCollection
            query = HttpContext.Current.Request.QueryString
            ' Will hold the name of each query string parameter
            Dim paramName As String
            ' Will host the value of each query string parameter
            Dim paramValue As String
            ' Used to parse the query string
            Dim i As Integer
            ' Parse every element of the query string
            For i = 0 To query.Count - 1
                ' We guard against null (Nothing) parameters
                If Not query.AllKeys(i) Is Nothing Then
                    ' Get the parameter name
                    paramName = query.AllKeys(i).ToString()
                    ' Test to see if the parameter is not ViewCart
                    If paramName.ToUpper <> "VIEWCART" Then
                        ' Get the value of the parameter
                        paramValue = query.Item(i)
                        ' Append the parameter to the page
                        newQueryString = newQueryString + paramName + "=" + paramValue +
"&"
                    End If
                End If
            Next
        Catch ex As Exception
            ' If something goes wrong we redirect to the main page
            Return String.Empty
        End Try
        Return newQueryString
    End Function

    Public Shared Function CleanShoppingCart(ByVal days As Integer)
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("CleanShoppingCart", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@Days", SqlDbType.SmallInt)
        command.Parameters("@Days").Value = days
        ' Open the connection, execute the command, close the connection
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Function

    Public Shared Function CountOldShoppingCartElements(ByVal days As Integer) As Integer
        ' Create the connection object
```

```
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("CountOldShoppingCartElements", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@Days", SqlDbType.SmallInt)
        command.Parameters("@Days").Value = days
        ' Execute the command and return the result
        Dim count As Integer = 0
        Try
            ' Execute the command and get the result
            connection.Open()
            count = command.ExecuteScalar()
        Finally
            ' Close the connection
            connection.Close()
        End Try
        ' Return the count value
        Return count
    End Function
    Public Shared Function CreateOrder() As String
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("CreateOrder", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@CartID", SqlDbType.Char, 36)
        command.Parameters("@CartID").Value = shoppingCartId
        ' Save the value that needs to be returned to a variable
        Dim orderId As String
        ' Execute the command and return the result
        Try
            ' Try to execute the command
            connection.Open()
            orderId = command.ExecuteScalar()
        Catch e As Exception
            ' Close the connection and rethrow the exception
            connection.Close()
            Throw e
        Finally
            ' Close the connection
            connection.Close()
        End Try
        ' Return the saved value
        Return orderId
    End Function
End Class
```

OrderManager.vb

```
Imports System.Data.SqlClient

Public Class OrderInfo
    Public OrderID As String
    Public TotalAmount As Decimal
    Public DateCreated As String
    Public DateShipped As String
    Public Verified As Boolean
    Public Completed As Boolean
    Public Canceled As Boolean
    Public Comments As String
    Public CustomerName As String
    Public ShippingAddress As String
    Public CustomerEmail As String
End Class

Public Class OrderManager
    Private Shared ReadOnly Property connectionString() As String
        Get
            Return ConfigurationSettings.AppSettings("ConnectionString")
        End Get
    End Property

    Public Shared Function GetMostRecentOrders(ByVal count As Integer) As SqlDataReader
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
        Dim command As New SqlCommand("GetMostRecentOrders", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and supply a value for it
        command.Parameters.Add("@Count", SqlDbType.SmallInt)
        command.Parameters("@Count").Value = count
        ' Return the results
        Try
            ' Open the connection
            connection.Open()
            ' Return an SqlDataReader to the calling function
            Return command.ExecuteReader(CommandBehavior.CloseConnection)
        Catch e As Exception
            ' Close the connection and rethrow the exception
            connection.Close()
            Throw e
        End Try
    End Function

    Public Shared Function GetOrdersBetweenDates(ByVal startDate As String, ByVal endDate
As String) As SqlDataReader
        ' Create the connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the command object
```

```
    Dim command As New SqlCommand("GetOrdersBetweenDates", connection)
    command.CommandType = CommandType.StoredProcedure
    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@StartDate", SqlDbType.SmallDateTime)
    command.Parameters("@StartDate").Value = startDate
    ' Add an input parameter and supply a value for it
    command.Parameters.Add("@EndDate", SqlDbType.SmallDateTime)
    command.Parameters("@EndDate").Value = endDate
    ' Return the results
    Try
        ' Open the connection
        connection.Open()
        ' Return an SqlDataReader to the calling function
        Return command.ExecuteReader(CommandBehavior.CloseConnection)
    Catch e As Exception
        ' Close the connection and rethrow the exception
        connection.Close()
        Throw e
    End Try
End Function

Public Shared Function GetUnverifiedUncanceledOrders() As SqlDataReader
    ' Create the connection object
    Dim connection As New SqlConnection(connectionString)
    ' Create and initialize the command object
    Dim command As New SqlCommand("GetUnverifiedUncanceledOrders", connection)
    command.CommandType = CommandType.StoredProcedure
    ' Return the results
    Try
        ' Open the connection
        connection.Open()
        ' Return an SqlDataReader to the calling function
        Return command.ExecuteReader(CommandBehavior.CloseConnection)
    Catch e As Exception
        ' Close the connection and rethrow the exception
        connection.Close()
        Throw e
    End Try
End Function

Public Shared Function GetVerifiedUncompletedOrders() As SqlDataReader
    ' Create the connection object
    Dim connection As New SqlConnection(connectionString)
    ' Create and initialize the command object
    Dim command As New SqlCommand("GetVerifiedUncompletedOrders", connection)
    command.CommandType = CommandType.StoredProcedure
    ' Return the results
    Try
        ' Open the connection
        connection.Open()
        ' Return an SqlDataReader to the calling function
        Return command.ExecuteReader(CommandBehavior.CloseConnection)
```

```
    Catch e As Exception
        ' Close the connection and rethrow the exception
        connection.Close()
        Throw e
    End Try
End Function

Public Shared Function GetOrderInfo(ByVal orderId As String) As OrderInfo
    ' Create the Connection object
    Dim connection As New SqlConnection(connectionString)
    ' Create and initialize the Command object
    Dim command As New SqlCommand("GetOrderInfo", connection)
    command.CommandType = CommandType.StoredProcedure
    ' Add an input parameter and set a value for it
    command.Parameters.Add("@OrderID", SqlDbType.Int)
    command.Parameters("@OrderID").Value = orderId
    ' The SqlDataReader object used to get the results
    Dim reader As SqlDataReader
    ' Get the results
    Try
        ' Open the connection
        connection.Open()
        ' Return an SqlDataReader to the calling function
        reader = command.ExecuteReader(CommandBehavior.CloseConnection)
    Catch e As Exception
        ' Close the connection and throw the exception
        connection.Close()
        Throw e
    End Try
    ' We move to the first (and only) record in the reader object
    ' and store the information in an OrderInfo object.
    Dim orderInfo As New OrderInfo()
    If reader.Read() Then ' returns true if there are records
        orderInfo.OrderID = reader("OrderID").ToString()
        orderInfo.TotalAmount = reader("TotalAmount").ToString()
        orderInfo.DateCreated = reader("DateCreated").ToString()
        orderInfo.DateShipped = reader("DateShipped").ToString()
        orderInfo.Verified = Boolean.Parse(reader("Verified").ToString())
        orderInfo.Completed = Boolean.Parse(reader("Completed").ToString())
        orderInfo.Canceled = Boolean.Parse(reader("Canceled").ToString())
        orderInfo.Comments = reader("Comments").ToString()
        orderInfo.CustomerName = reader("CustomerName").ToString()
        orderInfo.ShippingAddress = reader("ShippingAddress").ToString()
        orderInfo.CustomerEmail = reader("CustomerEmail").ToString()
        ' Close the reader and its connection
        reader.Close()
        connection.Close()
    End If
    ' Return the information in the form of an OrderInfo object
    Return orderInfo
End Function
```

```
Public Shared Function GetOrderDetails(ByVal orderId As String) As SqlDataReader
    ' Create the Connection object
    Dim connection As New SqlConnection(connectionString)
    ' Create and initialize the Command object
    Dim command As New SqlCommand("GetOrderDetails", connection)
    command.CommandType = CommandType.StoredProcedure
    ' Add an input parameter and set a value for it
    command.Parameters.Add("@OrderID", SqlDbType.Int)
    command.Parameters("@OrderID").Value = orderId
    ' Get the results
    Try
        ' Open the connection
        connection.Open()
        ' Return an SqlDataReader to the calling function
        Return command.ExecuteReader(CommandBehavior.CloseConnection)
    Catch e As Exception
        ' Close the connection and throw the exception
        connection.Close()
        Throw e
    End Try
End Function

Public Shared Sub UpdateOrder(ByVal orderInfo As OrderInfo)
    ' Create the Connection object
    Dim connection As New SqlConnection(connectionString)
    ' Create and initialize the Command object
    Dim command As New SqlCommand("UpdateOrder", connection)
    command.CommandType = CommandType.StoredProcedure
    ' Add the @Verified parameter
    command.Parameters.Add("@Verified", SqlDbType.Bit)
    If orderInfo.Verified Then
        command.Parameters("@Verified").Value = 1
    Else
        command.Parameters("@Verified").Value = 0
    End If
    ' Add the @Completed parameter
    command.Parameters.Add("@Completed", SqlDbType.Bit)
    If orderInfo.Completed Then
        command.Parameters("@Completed").Value = 1
    Else
        command.Parameters("@Completed").Value = 0
    End If
    ' Add the @Canceled parameter
    command.Parameters.Add("@Canceled", SqlDbType.Bit)
    If orderInfo.Canceled Then
        command.Parameters("@Canceled").Value = 1
    Else
        command.Parameters("@Canceled").Value = 0
    End If
    ' Add the @OrderID parameter and set its value
    command.Parameters.Add("@OrderID", SqlDbType.Int)
    command.Parameters("@OrderID").Value = orderInfo.OrderID
```

```
        ' Add the @DateCreated parameter and set its value
        command.Parameters.Add("@DateCreated", SqlDbType.SmallDateTime)
        command.Parameters("@DateCreated").Value = orderInfo.DateCreated
        ' @DateShipped will be sent only if the user typed a date in that
        ' text box; otherwise we don't send this parameter, as its default
        ' value in the stored procedure is NULL
        If orderInfo.DateShipped.Trim <> "" Then
            command.Parameters.Add("@DateShipped", SqlDbType.SmallDateTime)
            command.Parameters("@DateShipped").Value = orderInfo.DateShipped
        End If
        ' Add the @Comments parameter and set its value
        command.Parameters.Add("@Comments", SqlDbType.VarChar, 200)
        command.Parameters("@Comments").Value = orderInfo.Comments
        ' Add the @CustomerName parameter and set its value
        command.Parameters.Add("@CustomerName", SqlDbType.VarChar, 50)
        command.Parameters("@CustomerName").Value = orderInfo.CustomerName
        ' Add the @ShippingAddress parameter and set its value
        command.Parameters.Add("@ShippingAddress", SqlDbType.VarChar, 200)
        command.Parameters("@ShippingAddress").Value = orderInfo.ShippingAddress
        ' Add the @CustomerEmail parameter and set its value
        command.Parameters.Add("@CustomerEmail", SqlDbType.VarChar, 50)
        command.Parameters("@CustomerEmail").Value = orderInfo.CustomerEmail
        ' Execute the command, making sure the connection gets closed
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Sub

    Public Shared Sub MarkOrderAsVerified(ByVal orderId As String)
        ' Create the Connection object
        Dim connection As New SqlConnection(connectionString)
        ' Create and initialize the Command object
        Dim command As New SqlCommand("MarkOrderAsVerified", connection)
        command.CommandType = CommandType.StoredProcedure
        ' Add an input parameter and set a value for it
        command.Parameters.Add("@OrderID", SqlDbType.Int)
        command.Parameters("@OrderID").Value = orderId
        ' Execute the command, making sure the connection gets closed
        Try
            connection.Open()
            command.ExecuteNonQuery()
        Finally
            connection.Close()
        End Try
    End Sub

    Public Shared Sub MarkOrderAsCompleted(ByVal orderId As String)
        ' Create the Connection object
        Dim connection As New SqlConnection(connectionString)
```

```
    ' Create and initialize the Command object
    Dim command As New SqlCommand("MarkOrderAsCompleted", connection)
    command.CommandType = CommandType.StoredProcedure
    ' Add an input parameter and set a value for it
    command.Parameters.Add("@OrderID", SqlDbType.Int)
    command.Parameters("@OrderID").Value = orderId
    ' Execute the command, making sure the connection gets closed
    Try
        connection.Open()
        command.ExecuteNonQuery()
    Finally
        connection.Close()
    End Try
End Sub

Public Shared Sub MarkOrderAsCanceled(ByVal orderId As String)
    ' Create the Connection object
    Dim connection As New SqlConnection(connectionString)
    ' Create and initialize the Command object
    Dim command As New SqlCommand("MarkOrderAsCanceled", connection)
    command.CommandType = CommandType.StoredProcedure
    ' Add an input parameter and set a value for it
    command.Parameters.Add("@OrderID", SqlDbType.Int)
    command.Parameters("@OrderID").Value = orderId
    ' Execute the command, making sure the connection gets closed
    Try
        connection.Open()
        command.ExecuteNonQuery()
    Finally
        connection.Close()
    End Try
End Sub

    End Class
```

APPENDIX C

ACRONYMS AND ABBREVIATIONS

ASP .NET	ASP .NET is a set of technologiesdeveloped by Microsoft for building Web applications and XML Web Services. ASP .NET pages execute on the server and generate markup such as HTML, WML or XML that is sent to a desktop or mobile browser.
GUI	Graphical User Interface. The graphicalrepresentation of physical or pseudo-physical objects (such as buttons, trees, and lists) that allow the user to direct the flow of the program through the use of a mouse or other pointing device.
HTML	Hypertext Markup Language. A language that describes the formatting of text inside a browser.
Microsoft SQL Server	Microsoft SQL Server is a relational database management and analysis system for e-commerce, line-of-business, and data warehousing solutions.
.NET	Microsoft .NET is a set of Microsoft software technologies for connecting your world of information, people, systems, and devices. It enables an unprecedented level of software integration through the use of XML Web services: small, discrete, building-block applications that connect to each other—as well as to other, larger

	applications—via the Internet.
Visual Basic .NET	A newest Visual Basic tool set developed by Microsoft that enables developers to create rich applications for Microsoft Windows® in less time, incorporate data access from a wider range of database scenarios, create components with minimal code, and build Web-based applications using the skills they already have.

REFERENCES

[1] WHAT IS .NET
http://www.microsoft.com/net/Basics.mspx

[2] What is the .NET Framework, Micrsoft .NET Framework Developer Center
http://msdn.microsoft.com/netframework/technologyinfo/overview

[3] ASP .NET Architecture, NET Framework Developer's Guide
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconaspnetarchitecture.asp

[4] Siler, B. and Spotts, J., Using Microsoft Visual Basic .NET, QUE 2002.

[5] Evjan, B., et al., Visual Basic .NET Bible, Hungry Minds, 2002.

[6] The ADO .NET Data Architecture
http://www.startvbdotnet.com/ado/default.aspx

[7] Walther, S. and Levine, J., Teach yourself E-Commerce Programming with ASP, Sams, 2000.

[8] IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1993).

[9] Kauffman, J. and Claudio, F., et al., Beginning ASP .NET Databases Using VB .NET, Wrox 2003.

[10] Computer Center Web Application Development Standards, Westen California University, Pages 5 - 9.

[11] Fowler, M., UML distilled: a brief guide to the standard object modeling language, second edition, Addison Welsey Longman Inc., 1999.

[12] Microsoft Visual Studio .NET 2002
http://msdn.microsoft.com/vbasic

[13] Guthrie, S., Performance Tuning and ASP .NET Whidbey http://weblogs.ASP .NET/scottgu/archive/2003/11/18/38480.aspx

[14] Hull, S., PHP and ASP .NET Go Head-to-Head http://www.oracle.com/technology/pub/articles/hull_asp.html

[15] Debate - .NET V. PHP: Top 10 .NET Myths Exposed http://www.sitepoint.com/article/php-top-10-net-myths-exposed/20.

[16] Scott, The ASP .NET Web Matrix Project Reloaded http://aspnet.4guysfromrolla.com/articles/061803-1.aspx

[17] Johnsson, M., Debate - .NET V. PHP: Top 6 Reasons to Use .NET http://www.sitepoint.com/article/v-php-top-6-reasons-use-net